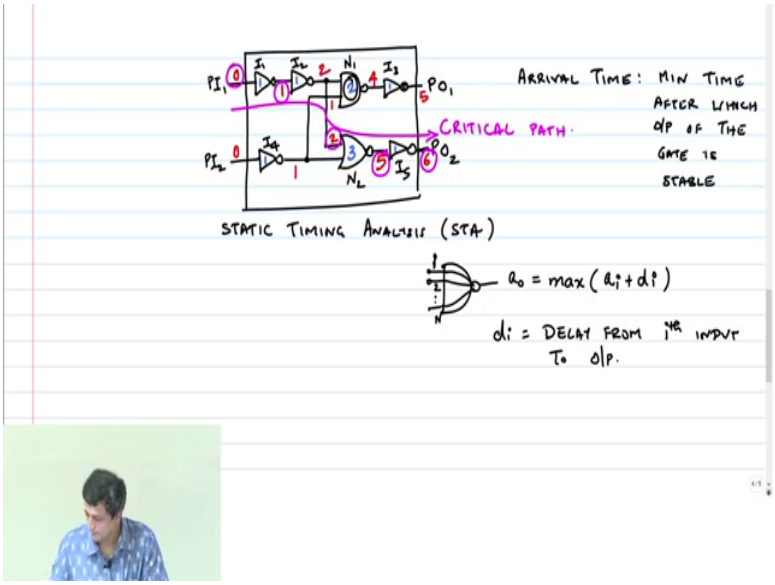
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 53
Gate Sizing for Large Circuits

Now, I am going to take all that we have learnt and put it together and see how we can apply this to a real world circuits. What is it that is given to us in the real world and how do we apply this concepts that we have learned in this module to solve the; those real world problems ok.

(Refer Slide Time: 00:36)



So, let me consider a very simple example first. So, first we will do what is known as static timing analysis; timing analysis is something that you do on a circuit given that the sizing has been done already; which means that the delay of the gate is fixed ok, we will solve the sizing

problem next. So, let us assume that we have done the sizing and I will indicate the delay inside the gate. An inverter has 1 unit delay, the NAND has 2 units delay ok, the NOR; I am saying has 3 units of delay; let us say. You have sized it like that and therefore, the delay is in that version ok.

So, the terminology is; this is your black box circuit that you have, these are my primary inputs PI 1, PI 2; this is primary output 1, primary output 2 ok. We define something known as an arrival time; arrival time is basically the time at which the output of the gate will be stable; the minimum time after which the output of the gate will be stable ok.

So, you could have some transitions and then it will stabilizes to that output that is what we are calling arrival time; it is min time, after which output of the gate ok. So, if I have a logic gate with say N inputs; if I ask you when will the output of this gate be stable? The minimum time after which it will be stable; well, obviously, it depends on when the inputs arrived and when the inputs were stabled right if the inputs which is after some time of course, output do will also change.

So, the output arrival time is indeed related to the input arrival time and in general if I have delay from one input to the output all different right. So, if I have input i; no 1, 2 you know N and my output is just one of them, then let dij equal to delay from ith input to output right. I can say; I can remove the j, typically in whatever I have shown here this 2 is the same for both inputs. It does not matter which input to output we are looking at the delay is only 2. In most cases this di is similar ok, but the arrival time at the output of this gate therefore, will be what?

Student: Maximum.

Maximum of; Maximum of? Arrival time at the ith input plus the delay from the ith input to the jth to the output correct.

So, this is arrival time at the ith input plus di ok; we will sense a; small ao; clearly there is a maximum operation that we have to perform here right. By definition arrival time at primary

inputs is 0; they arrival time 0 right, it is just a definition. So, arrival time here ok; I am going to mark it in red is 0. What is your arrival time at the output of the first inverter? Let me label them I 1, I 2, NAND gate; N 1, I 3, I 4, N 2 and I 5; what is the arrival time at the output of I 1?

Student: (Refer Time: 05:49).

0 plus the delay through the inverter; therefore, you get arrival time of 1. What about arrival time at I 2? It will be 2. What about arrival time at N 1? Be careful here with depends on when the other guy came; depends on when the other guy comes. So, therefore, you call to first solve for I 4. What is the arrival time at the output for I 4? 1. Now, therefore the arrival at this node is also 1, now you have to do max of arrival time at first input which is 2 plus 2 and 1 plus 2. And therefore, this will be 4; what is the arrival time at the output of I 3?

Student: 5.

5 ok; now, so therefore the arrival time here is 2; what is the arrival time at the output of N 2?

Student: (Refer Time: 06:54).

Yeah, you have to do 2 plus 3, 2 plus 3 and take max; so you get 5. What is the arrival time at the primary output? PO 2; I mean yeah at PO 2.

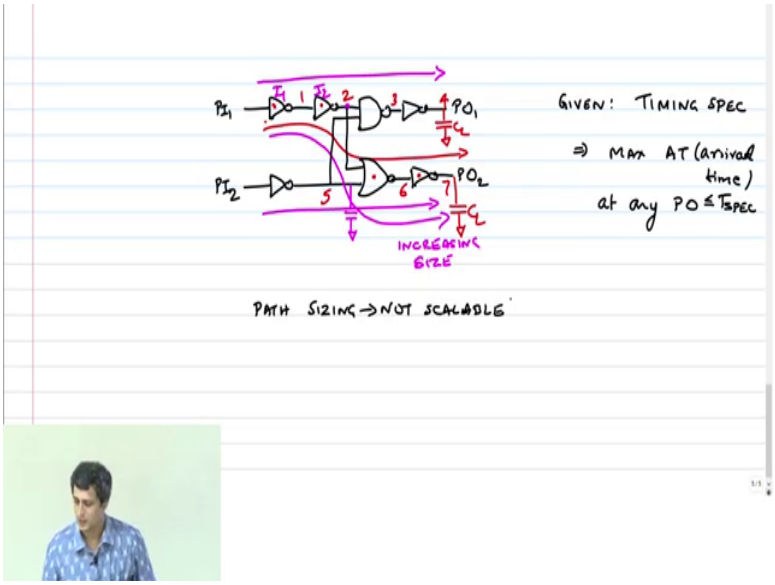
Student: 6.

6 ok. So, now I can define what is known as a critical path; the critical path is the path from input to output which results in maximum time for the input to; for any input to propagate to the output; that means, after the critical path time, every output will be stable ok. So, what is now how do I calculate the; how do I find the critical path? I take the maximum arrival time and then back propagate right.

So, to calculate the critical path; I will take 6, then I come one step back; it is 5; now for the NOR gate, which side should I go? Should I go towards I 4 or I 2? I 2. Again take the max there, go back this guy; then I go back here and go here. Therefore, the critical path is like this right; so that is called the; this is called the critical path ok.

So, now I am going to flip the problem; I gave you the sizes, I gave you the delays; we could find a critical path, we could do some analysis. Suppose I did not do that right; I had to now do my gate sizing problem on this circuit. What is the gate sizing problem on this circuit? I need to find the input capacitance of every gate so that the latest arrival time at any of the outputs does not exceed a timings spec; this is what I need to do right.

(Refer Slide Time: 09:18)



So, I am going to flip this now; then PI 1, PI 2, PO 1, PO 2; you are now given a timings spec which means that the maximum arrival time right meaning is max AT ok. I am going to call

this arrival time AT at any primary output should be less than or equal to T spec right; only then I can guarantee that the circuit will function at this particular speed. There is no input combination that can cost the output to be stable after this T spec; that is what we have done here remember. So, what is this? This analysis by the way that we did earlier is called static timing analysis.

Here, what are we doing? We are considering all possible combination of input variations and ensuring that the output in the worst case does not exceeds some particular time. So, here for example, it might be possible that a particular path that we are marked here may not we in be realize; realizable in hardware. In the sense, not realizable that path we may never get activated because of certain input combinations, but still we are considering the worst case delay there; you get it? So, if I have a NAND gate with the input always tie to 0, the output will be 1.

The inputs switching on the other thing is not going to affect the output right. Therefore, if I say that you know the output is of that NAND gate is going to depend on the input to which is connect; which is always connected to 0, it does not make sense. But we are still considering those as well; worst case timing analysis is what we are doing in static timing analysis this is called STA ok.

So, now using the previous example and whatever we have learnt in this course; what is it that you would do in order to do the sizing problem? So, what did we learn? We have learned how to optimize the path delay of a given path; the delay of a path you know how to do.

Now, clearly because there are maximum number of gates in this path; we might say ok, let us be that path and optimize the delay of that path right; it is a fair thing to do. Because that is going to give you the largest delay and if you ensure that the largest delay is less than T spec; then you have fine, but unfortunately you saw in the previous example that the other output arrival time was not far behind; it was only 5 right.

So, let us see what happens if I apply this path delay sizing problem to this circuit. What will I do? I will now start sizing this particular path right I 1, I 2; let me just make sure by notation

is correct because I can switch some slides to this has some part right. So, that my path ok; yeah let me not call this I 1, I 2; I am going to call this 1, 2, 3, 4, 5, 6 and 7 ok.

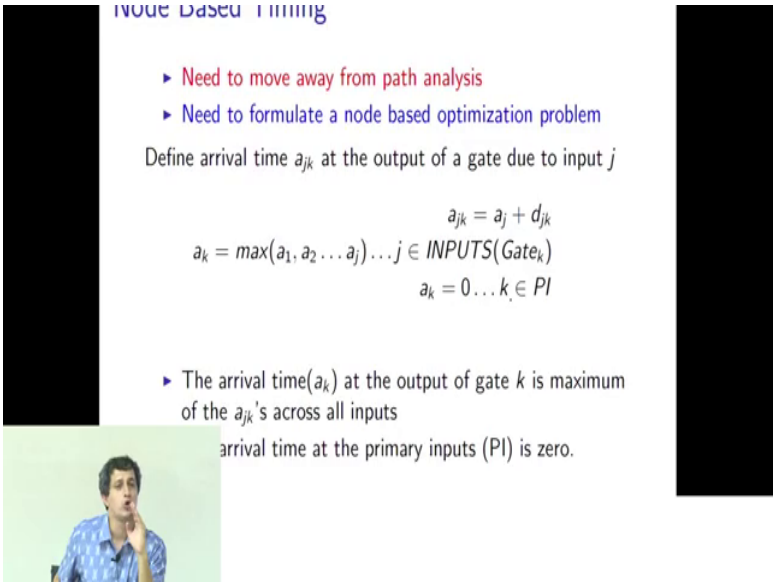
The output of course, there is a load capacitor that has to drive some load capacitance. So, what I will do? I will now pick this inverter; this inverter; this NOR gate this inverter and do my path delay sizing. In the process, what are you going to do? You will end up upsizing the gate progressively from input to output; the gate sizes will start going up or the capacitance start going up as a by what factor? Stage effort; it goes up by the factor of the stage effort right. Therefore, this will be increasing size; in the process, what you will do is; you will increase the capacitance on this node.

Now, if you increase a capacitance on this node; you are going to increase a capacitance of this node as well right; on node 2. Capacitance on node 2 will go up which means now that this path delay may go up. Likewise this path delay may also go up because that NAND gate right; the wherever the; no not that path delay actually this cannot go up it is only the other path delay that can go up right whichever is connected to that inverter I 1, I 2. Since a NOR is connected to I 2 that delay can go up; similarly this delay can go up right.

In effect what we have done is; we have in order to optimize the delay of one particular path, we have ended up increasing the delay of the other path. And there is no guarantee that in this process of gate sizing what we identified as a critical path initially will remain a critical path always. It can easily switch at any point; effectively saying you cannot do path delay base sizing on an actual circuit, simply; because I have exponential number of paths; exponential in what? In the number of modes.

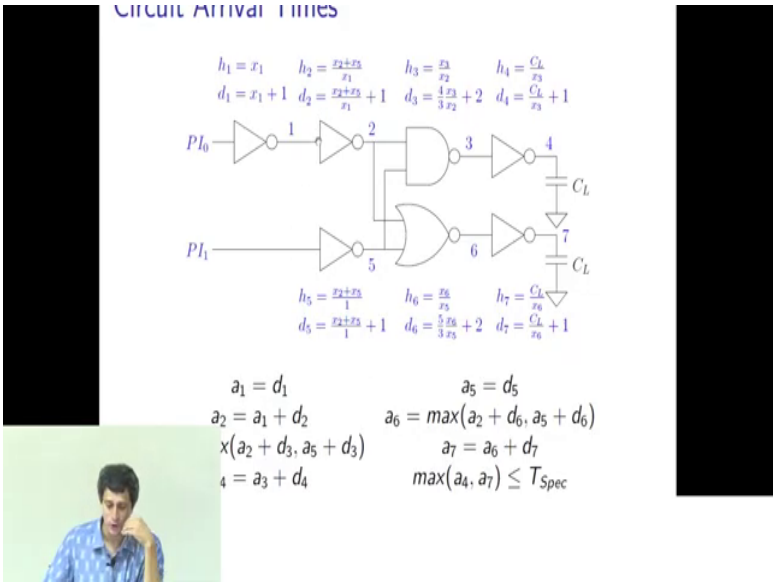
So, if I have N gates or N nodes in a circuit; then this is O of you know exponential of; the number of paths. And there is no way I can optimize all the paths simultaneously because I cannot even enumerate all the paths, forget about sizing all the paths right. So, first thing is path sizing not scalable ok; in the interest of time, I will just switch to the slides right. So, that you just get the quick feel; I want to complete this way.

(Refer Slide Time: 17:21)



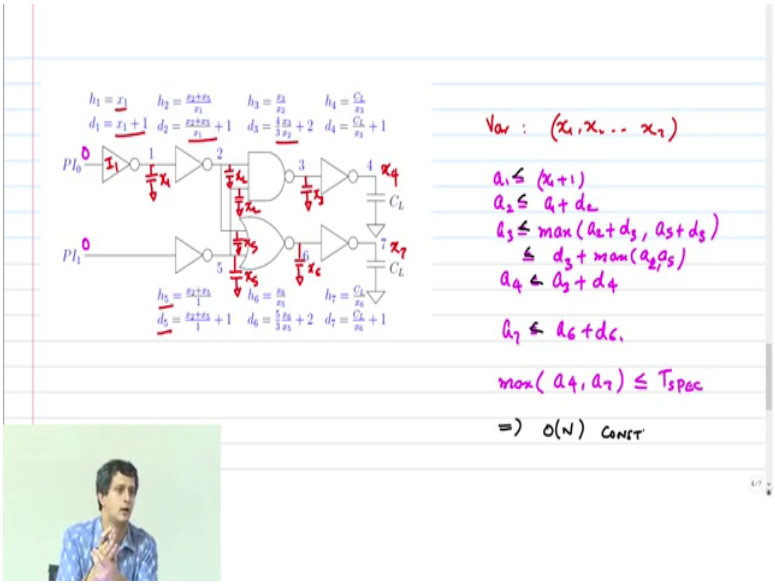
So, what I am going to do now is I am going to show you how people move away from a path based formulation to a node based formulation; that means, if you have N nodes; the number of equations and other things should be O of; then it is it is solvable of course, if you have large security it take long time to solve, but it should be in still in the order of number of nodes of a circuit right. So, this we did; arrival time at the kth output is basically max of arrival time of the input you know and all that stuff right.

(Refer Slide Time: 17:56)



So, let us assume that the output the capacitance of node 1 is xk right. So, let me just copy this ok.

(Refer Slide Time: 18:24)



So, what we are saying is the capacitance on the gate whose output node is k; the capacitance of the input is xk, this is going to be x 6; this capacitance; similarly this capacitance here will be what? x 3.

So, this is basically going to be x 1; is it oh no; that cannot to be the case, x 3 is x 3 oh no sorry no. This one is x 3; this is x 2, x 2; x 5, x 5 right and this is basically x 6; this is x 4, x 7, x 1 ok. These are my variables; my variables are as usual are my capacitance x 1, x 2 all the way to x 7; I have to solve for this ok. So, what is the delay of inverter I 1? It is g h plus p; what is h?

Student: (Refer Time: 20:11).

Yeah?

Student: (Refer Time: 20:13).

x 1 right; assuming that the first inverter is the one x inverter ok; let us keep that as an assumption. So, h 1 is basically just x 1 and therefore, the delay of the inverter is x 1 plus 1 right, g is 1. What about h 2? It is now the net capacitance on node 2 is x 2 plus x 5 right because the same node inverter is driving the NAND and the NOR. So, it is x 2 plus x 5 by x 1 that is h and therefore, the delay is x 2 plus x 5 by x 1 plus 1.

Similarly, for the NAND 3; it is simply x 3 by x 2; remember it is not x 3 by x 2 plus x;5 it is only the input capacitance of that particular thing right. So, it is x 3 by x 2 and the delay is logical effort 4 by 3 into x 3 by x 2 plus 2. Similarly, I can right down the delay for all the other gates right just that for example, in h 5; your I mean in d 5, you have to be careful that this inverter is driving the NAND and the NOR again.

So, therefore, again it is x 2 plus x 5 by 1; plus 1, that is the delay; clear? This is something nothing new to us; whatever we learned earlier is just what we are applied here. Only thing is now I have to take this and alter my optimization ok. So, what is now the arrival time at various nodes? What is a 1; arrival time? Remember the arrival time at the input is 0 primary inputs. So, what is a arrival time at node 1?

Student: (Refer Time: 22:16).

Yeah?

Student: (Refer Time: 22:18).

x 1 plus 1 right; a 1, what is a 2? It is a 1 plus d 2 right; what is a 3?

Student: (Refer Time: 22:44).

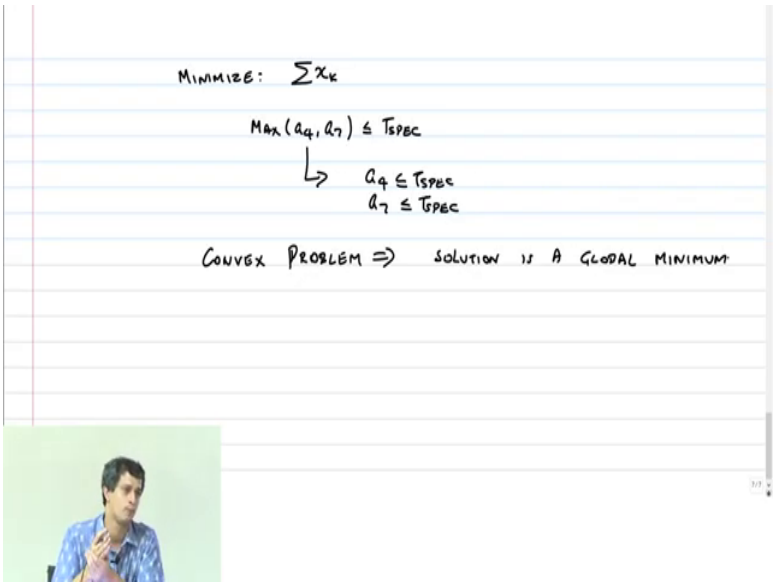
Huh.

Student: (Refer Time: 22:47).

Now, you should be careful max of a 2 plus d 3 comma a 5 plus d 3 ok. This I can remove d 3 common; d 3 plus max of a 2 comma; a 2 comma a 5 ok; a 4 I can write similarly in a similar way right which is basically arrival time at a 3 plus d 4 right. I can now right down all these things. So, effectively I have a 7 eventually will be what? Arrival time at a 6 plus d 6 right.

What is the constraint that I have on my circuit; given a timing spec, max of a 4 comma a 7 should be less than or equal to T spec right. This is what I have to now set up how I have to set up the optimization problem ok. Now, typically you do not try to minimize the delay of a circuit because you are given a timing spec; if you try to minimize the delay you will make this circuit much faster than the spec, but end up burning more power or more area. So, therefore, what I am going to do is; I am going to set up my optimization problem to be the following.

(Refer Slide Time: 24:24)



I will minimize summation xk; what is summation xk? If I minimize summation xk, what am I minimizing here? Area or power; equivalently it is power right, it is the; though there is not exactly the same you will minimize both in the process; such that max of a 4 comma a 7 is less than or equal to T spec. Now, the maximum operation is very painful to do an optimization. So, therefore I have to make this a separate some other kind of you know condition.

So, if I just simply break this as a 4 should be less than or equal to T spec and a 7 should be less than or equal to T spec; then I can guarantee that the max of this is less than T spec right. The other problem is the equality condition in optimization is a big problem; therefore, what you do is you simply convert all of these to less than or equal to (Refer Time: 25:39).

It is effectively the same thing these are called auxiliary variables in optimization; you just insert this variables make all equality conditions with less than or equal to and solve the problem; you will get the same answer right. Now, what is the catch here? We have done everything; it turns it will it come out very well, you get a solution of you know x 1 to x 7. How are you sure that this is a minimum or this is a global minimum? You will get one answer, you use the optimization tool, you use put this constraints in; you minimize summation x 1 to xk; you will get one answer.

How are you sure that this is a really an optima; that means, is there some other sizing, some other setoff sizes which will give me lesser area for the same T spec? It turns out; no, this is a convex optimization problem and this class of problems is called a geometric programming problem ok.

So, this kind of an optimization where you have x by; x 1 by x 2 kind of terms it turns out its a convex; convex problem, implies solution is a global minimum right; that is a question you ask me long back saying in path delay how are you sure that its path delay is also the same thing? It is a geometric programming problem and the solution is guarantee to be an optimal cord.

Now, what is the big thing that we are done here? What we are done is just look at the number of constraints that we are introduced. It is just O of N; if I have N nodes in my circuit; how many node do I have here? 7 nodes, the number of constraints are basically 7 at you know for each input of the gate; I may have something. This is O of N constraints; there by even if you give me a circuit with a million gates, I can solve it first in reasonable time because the number of constraints is only O of N to I can guarantee that that is a global optimal; there is no other solution which will give your better delay or a better area for that given delay right.

So, this is how you take what we did in this module for toy examples of path delay and apply to actual circuits; your TDA tools which do timing optimization right; you do syntheses, you

get the logic circuit. Then you have to make sure that the logic satisfies timing; how does a do it? It does it exactly like this.