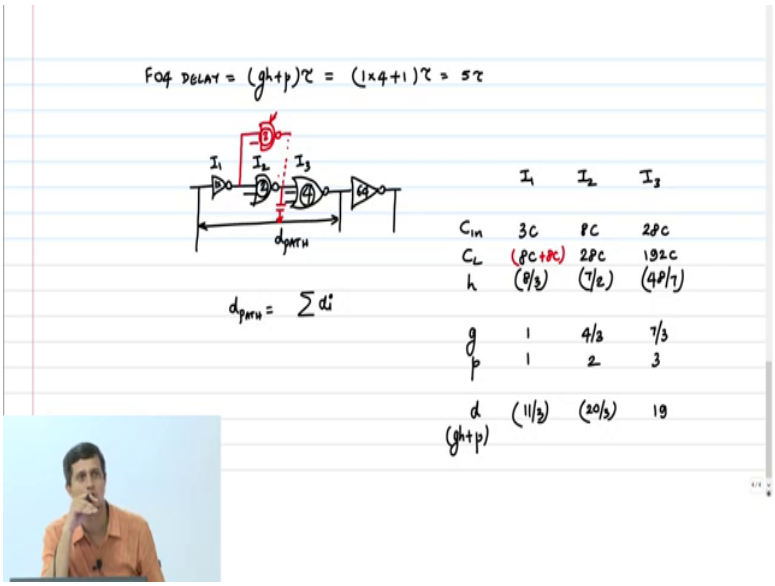
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture-42
Path Delay Calculation and Optimization Formulation

(Refer Slide Time: 00:15)



So, let us now take this expression that we derived, and just try to calculate the delay of a path, because next class I would like to start the path delay optimization ok. I am saying this is 1 x, 2 x, I will say 4 x, this is a NAND 2 this is NOR 3 and ultimately I will drive it by a capacitance which is 64 x ok. So, what I wanted to do is now calculate the delay of this path, d path sorry.

So, d is simply summation of the independent delays. There are four delays here that I have to calculate, I will not calculate up to the last one, sorry, I will calculate only up to here because

other I need otherwise I need to know the load there only three, three gates I am going to calculate ok. So, what we will do is we will write down for.

So, this is I 1, gate I 2 and I 3 ok. I am going to now write a large table and we have to fill out these numbers for I 1, I 2, I 3, C in C L load capacitance, logical effort, parasitic effort, no, I will may be say electrical effort first, then logical effort, parasitic effort finally, delay. So, if for I 1 what is the net input capacitance, how much?

Student: (Refer Time: 02:46).

3 C yes, right. What for I 2, what is the net input capacitance?

Student: (Refer Time: 02:56).

It is 4 C, but I have said it is a 2 x inverter, this two inside basically says it gives you the drive strength which means I have sized it up by a factor of 2 right. Therefore, the net input capacitance that it would see is how much.

Student: 8 C.

8 C. What about I 3? It is a NOR 3. So, how much do you see seven four, yeah, the NOR would normally present a capacitance of 7 C; 1 from NMOS, 6 from PMOS NOR 3. But I am upsizing by a factor of 4, and therefore I am getting a delay of how much, 28 C. What about the load capacitance for each of these things? I 1, it is?

Student: 8 C.

8 C right. I 2 28 C. What about for I 3, 192, 64 into 3 x into 3 C, you will get 192 C. So, what is the electrical effort h for each of these gates?

Student: (Refer Time: 04:32).

So, it is simply 8 by 3 for this guy. This is 28 by 8, 7 by 2, and I 3 is 192 by 28. What is that 48 by 7, 48 by 7 ok. Now, what is the logical effort? G for I 1 1; I 2 it is 4 by 3, I 3 NOR 3?

Student: (Refer Time: 05:27).

7 by, yes, parasitic delay effort 1 for inverter; for I 2 2. For I 3?

Student: (Refer Time: 05:44).

3 correct. So, can you now calculate the delay which is basically g h plus p? What is the answer for the inverter?

Student: (Refer Time: 06:02).

11 by 3, for I 2 NAND 2.

Student: 20 by 3.

20 by 3. And for NOR 3?

Student: 19, 19.

Yeah.

Student: 19.

19, ok. So, you add this and you will be able to get an estimate for the path delay ok. So, you have to be careful only in the following case what if I modified this circuit like this. This also

is 2 x. I add another NAND 2 parallely loading my inverter I 1 which of these numbers will now change?

Student: C l of I 1.

C l of.

Student: I 1.

I 1, this will become plus 8 C. What about C in of I 2, will that change or will that not change? So, that will not change that is the only thing you should be careful about because remember that this C in, we are actually trying to capture the effect of the output resistance that K that we got is how we wrote as C you know we wrote it as C du t by C unit.

Now, this gate is not contributing to charging this capacitance that is just loading the previous stage that is all; it does not contribute any current to charging the output load capacitance of gate I 2. And therefore, it should not be considered when you are calculating the C in of the gate under test. You can only consider that branching thing when there is a load that you want to calculate ok.

So, here the only thing that will change is the load capacitance of I 1 will become 16 C instead of this thing. Therefore, the h for I 1 will change; for I 2 the h will remain the same, ok, clear?

Student: (Refer Time: 08:59).

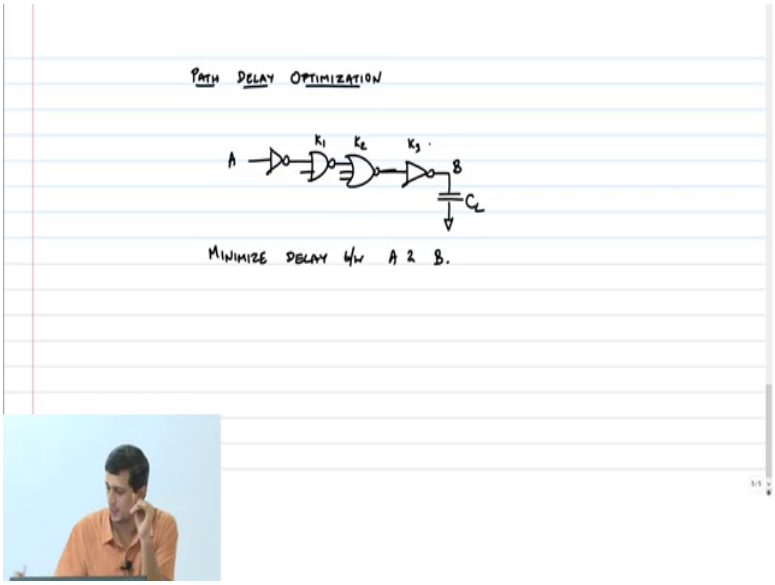
If it is connected.

Student: (Refer Time: 09:01).

You mean like this, then it is a different thing altogether that is a different case altogether; right now we are not considering this kind of a complicated circuit. I will tell you how people handle these circuits later on ok; it is not part of the syllabus, but just for academic purposes I will tell you how they do it ok. Basically you are saying there are multiple paths that are converging back to the same node that is far more complex; right now I am just talking about a single path.

You start from a source node, and then you can go out you never come back. Clear, any questions still now on logical effort on delay calculation, path delay calculation, because this is very important for us to proceed into the next part of this module. So, anyway, you do not have to worry because these concepts will get reinforced again and again we will keep coming back to this as we go on.

(Refer Slide Time: 10:27)

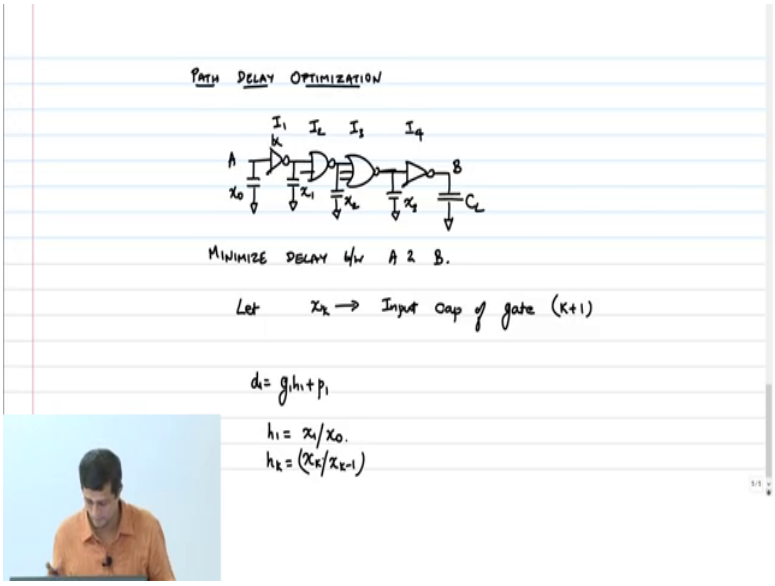


So, let us let me now start the most interesting problem that we would like to solve in digital circuit design right and that is basically called path delay optimization. In the previous example, I gave you the sizes I said this was 1 x that was 2 x 4 x and so on and said calculate the delay. Now, I am going to reverse the problem; I am going to tell you that there is a circuit that is sitting between an input node and an output node, I know the capacitance at the output node.

I want you to find the sizes, so that the delay between these two points will be minimum ok. So, it is like this. I have the same let me just take the same circuit ok. And let me put a, I will add one more now, there is a load capacitance ok. I want to find, I want to find the sizes of these gates which will minimize the delay between point A and point B, I want to minimize between A and B ok.

So, now, clearly if I went and chose my variables as let us say my variable was the size of the drive strength of the NAND gate let me call it K 1, this as K 2 this as K 3, suppose I did that, then you already saw in the previous case you have to put in a lot of effort to calculate the input and output capacitances, because it depends on the topology of the gate right, so that is not very useful, not very useful when slightly more cumbersome right. The delay expression we have derived is actually very very compact and nice if you choose your variables correctly.

(Refer Slide Time: 13:03)



So, instead of choosing the drive strength as my variables, I am going to choose the capacitance here load capacitance x 1, this capacitance x 2, and this capacitance x 3, sorry x 3 as my variable ok. So, let x k be the input cap. And of course, I need to assume that the first gate size is known, let us assume that is fixed that is the minimum size in gate, and this is just a reference inverter.

So, this is a 1 x gate. Let x k be the input cap of gate k plus 1 ok. This is the input gate gap of the next gate. So, you do not, make sure you do not count parasitic capacitances here, this is input gate gap of the next gate, it is just load capacitance, those are my variables ok.

So, now, what I can do is I can write my delay expression for each of these gates like we did earlier right. So, I have gate I 1, I 2, I 3 and I 4. So, I can write d 1 as g 1 into h 1 plus p 1. Of

course, here it is an inverter, so g 1 is 1, p 1 is 1 and so on. But in general I can write it as the delay of the kth stage is g k into h k plus p k ok. What is h 1 now?

Student: (Refer Time: 15:20).

Yeah.

Student: (Refer Time: 15:22).

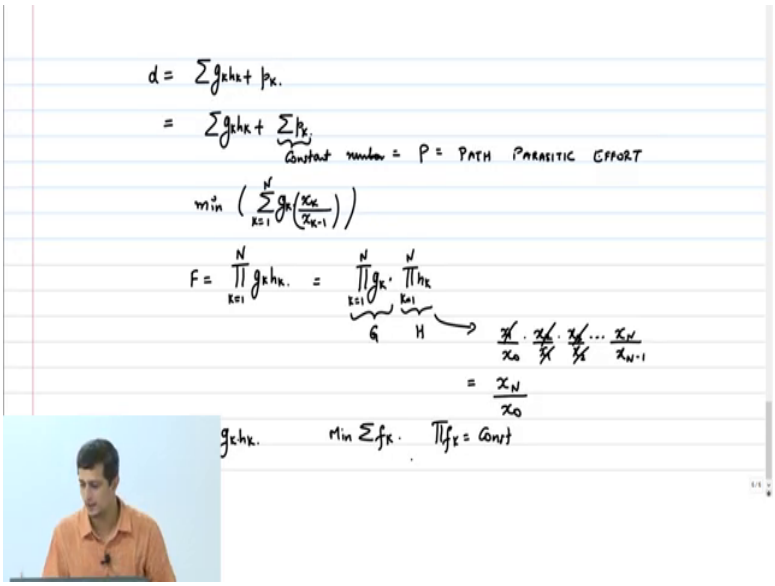
Sorry.

Student: (Refer Time: 15:24)

Student: (Refer Time: 15:26).

So, let us assume this is x naught, the input capacitance of the first gate is x naught. So, h 1 is x 1 by x naught. What about h k, therefore, x, it is just x k by k minus 1. This is the advantage if we choose the capacitance itself as the variable, then h is become very simple to evaluate, they are just ratio of these capacitances and I am done ok.

(Refer Slide Time: 16:23)



So, now, what do I want to do I want to find the following, I want to the net delay is summation g k h k plus p k. So, I will write this out as summation g k h k plus summation p k. Now, this summation p k is independent of my gate sizing, parasitic delay is independent of gate sizing. So, therefore, this is a constant number and I will call this as the path parasitic effort ok.

So, here is where I told you that it does not matter whether you do 7 r c or 6 r c or whatever it is for the NAND gate contamination or propagation or whatever, it is a constant number that is not affected by my gate sizing problem. So, when I do my minimization that constant anyway goes out, you want to be more accurate please be more accurate.

It does not affect your gate sizing algorithm is all I was trying to tell you there. And therefore, it is to make the approximation of only counting diffusion capacitances ok, so that is the

constant number. Now, what do I want to do, I want to minimize this objective function summation g k into x k by x k minus 1 right; k going from 1 to how many our gates there are in the circuit. I want to minimize this delay and my variables are x k s ok; x 0 is known; x n is known.

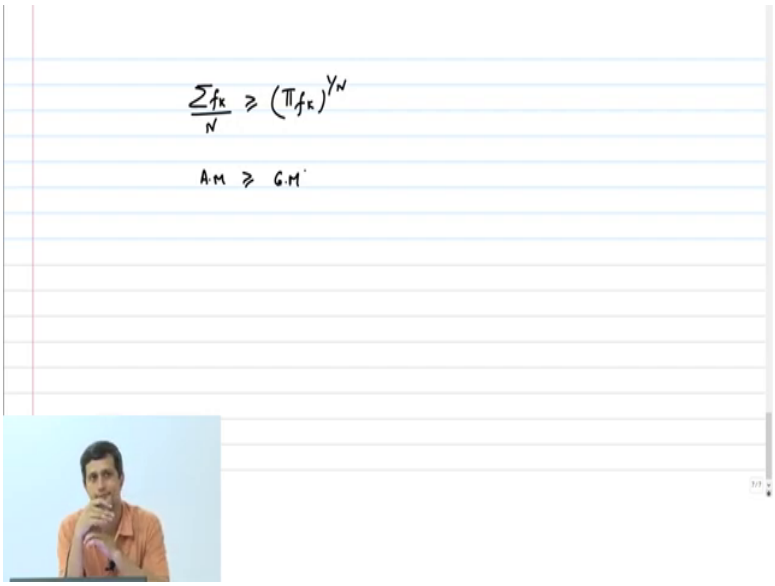
Now, let us look at the nature of this these terms ok. So, let me look at the product g k into h k. Just let me look at this term and I will call it a term f k equals 1 to N. This I can simply write as product of g k, 1 to N into product of h k, 1 to N. Now, this is some product of logical efforts ok. So, I will call this g, it is called path logical effort ok.

Now, this guy is my path electrical effort what is the path electrical effort, if you expand this and write it, it will simply become x 1 by x naught into x 2 by x 1 into x 3 by x 2 x n by x n minus 1. So, what happens all the variables simply cancel out, and you are only left with ratio of output load capacitance to input capacitance of that path, all the variables in between are gone ok.

So, therefore, this I will simply write I mean not sorry I can just simply keep canceling like this, and you will simply get as x n by x naught ok. And what is x n it is basically this, x n, my final load capacitance by input capacitance clear. Now you observe this term I have I want to minimize the sum of let each let f k equal to g k into h k. I want to minimize summation f k, but the property of f k is product of f k is a constant right.

So, what do you do now, you use arithmetic mean geometric mean inequality, and you can solve, the left hand side is some form of the arithmetic mean summation f k by N right.

(Refer Slide Time: 21:45)



So, I can say the following summation f k by N is greater than or equal to product of f k whole power 1 by N right. This is nothing but AM greater than or equal to GM inequality, and they will be mean the minimum will occur when each term is going to be equal ok. So, what we will do is we will stop here. This is the outline of the proof. Next time, we will complete this proof, solve a few examples, and then proceed with more interesting cases on the same lines.

Thank you.