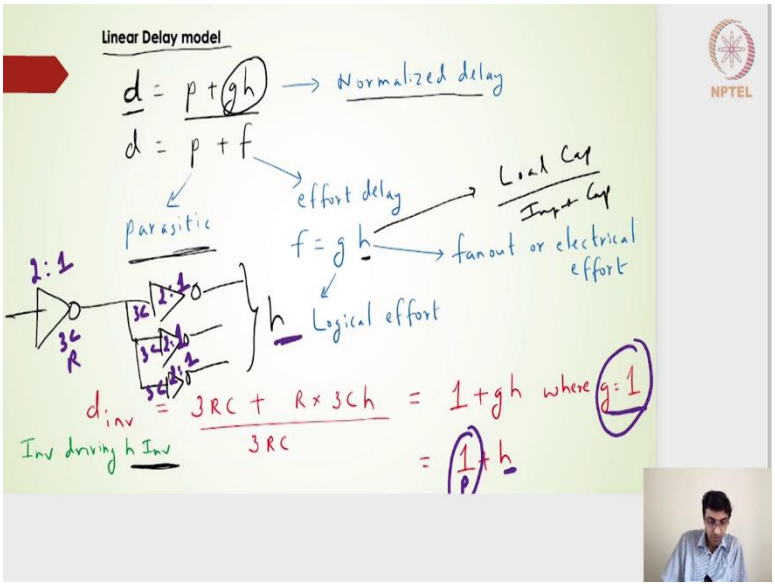
Design and Analysis of VLSI Subsystems
Dr. Madhav Rao
Department of Electronics and Communication Engineering
The International Institute of Information Technology, Bangalore

Lecture - 29
Linear Delay model

(Refer Slide Time: 00:16)



Hello students. Welcome back to this lecture on the delay of NAND and NOR gates. In this particular section we will look into more about p + gh model. We had seen an Elmore delay method and utilize that particular model or the Elmore delay method to estimate the delay maybe it is a falling delay or a rising delay. This particular model is also called as p + gh model and it is a linear delay model. It does not involve any, I mean it the delay whatever we estimate it will not be a non-linear function of the capacitance or the scale, it is a linear model and the overall delay, when I write the small lowercase d, it means that it is a normalized delay.

It means that it is the absolute delay been taken for a particular gate and then it has been divided by the inverters delay p + gh as we had seen in the last lecture and then this p represents that this is a parasitic delay and then this gh can also be considered as an f value which is nothing but the effort delay,

$$f = gh$$

The g is called as a logical effort and h is called the fanout of the electrical effort. In this particular definition of the fanout we had seen earlier,

$$\text{fanout} = \frac{\text{load capacitance}}{\text{input capacitance}}$$

What it means the h value is nothing but whatever is the input capacitances given by the next stage, that is our load capacitance and it is divided by the first stage or whatever the stage we are considering in this point of time its own input capacitance, that is the fan out of the electrical effort.

Logical effort we will see a little bit later and then the parasitic which is a normalized parasitic also the definition of it and then the expression for it we will also see in the future slides. If I am considering that inverter for this particular linear delay model, if I want to estimate the delay, the delay of the inverter is nothing but driving h inverters. I have one inverter in the first stage driving another inverter and h such inverter in the second stage, this is what the stage or the h represents here, I will get a 3RC.

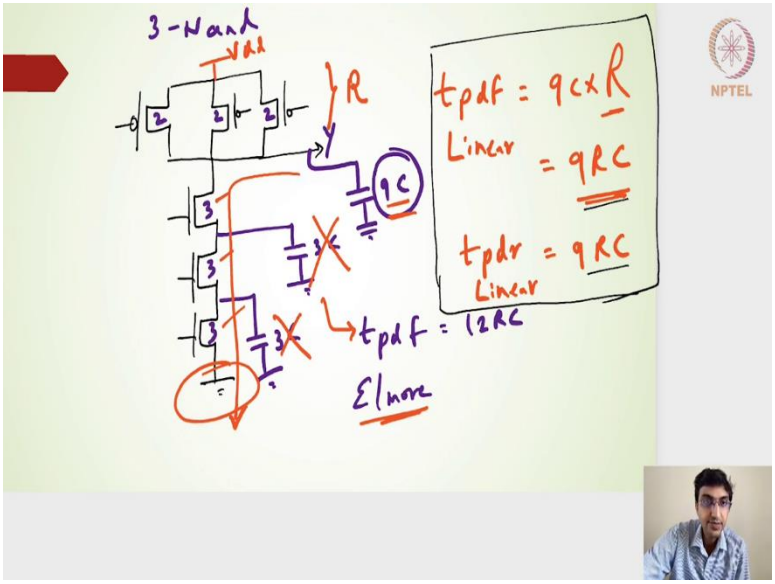
I am assuming this is 2:1 ratio, I will get each of this delay will give me the output node capacitance is 3C here, switching resistance is R. I will get 3RC and then plus the input capacitances here coming from here is 3C, 3C and then so on 3Ch is,

$$d_{inv} = \frac{3RC + Rx3CH}{3RC} = 1 + gh = 1 + h$$

Where p is nothing but 1, g is nothing but 1 and h is nothing but the value of h, how many number of inverters I am using it in the second stage or I am designing it in the second stage.

This is nothing but p is nothing but 1 here, g is nothing but 1 here and h is whatever is the number of inverters that we have designed for the second stage. Hope you have understood this particular linear delay model. I have an empty slide or that means that I have to draw something here.

(Refer Slide Time: 04:39)



Let us go back to our 3-input NAND gate and I am going to draw the 3-input NAND gate circuit again. This is my PMOS side and PMOS the input should be bubbled here and this is my output and then on the NMOS side I will have the 3 transistors in series, these are the inputs A, B, C.

What we had seen up till now is especially for using the Elmore delay method, we had used this inter node diffusion capacitance of 3C. Assuming that the sizes are 2, 2 and 2 here and then 3 here we get the capacitances here, the diffusion capacitances of 3C's. Then finally, the output node capacitance is $2 + 2 + 2 + 3 = 9C$.

This is what we had used and then using the switching resistance for each of these transistors we got the $t_{pdf} = 12RC$ and here I am going to write it as the Elmore delay method. The linear delay method has only one capacitance, it sees only it considers only the output node capacitance.

The capacitances which are associated at the output node and it does not consider any of this capacitances, the inter node diffusion capacitances. Whatever is the capacitances at the output node, it takes there into account the linear delay model and then the propagation delay falling. Using the linear delay model is nothing but 9C into whatever is the switching resistance and then the path that the switching resistance makes with respect to the sources.

In this case it is the source is the ground. In the rising one the source will be nothing but V_{dd}. In that case the equivalent switching resistance or the discharging path switching resistance will be R/3 here, R/3 here, R/3 here, it will be nothing but R. Finally, my value is nothing but 9RC.

The linear delay model does not account for the inter node diffusion capacitances, which was 3C and 3C in which we had use that to find out the propagation delay falling using the Elmore delay method. But, the linear delay model does not consider that and it considers only the capacitances that is connected at the output node, that is a major difference. The linear delay model is a much more approximation than that of the Elmore delay method.

But it is kind of very handy and then in the sense it is quite very fast method to arrive at the delay parameters. Given a circuit or a given schematic I should be able to see only the capacitance at the output node and then multiplied by whatever the switching resistance that gets connected to the ground or gets connected to the V_{dd} and that will be my the overall delay.

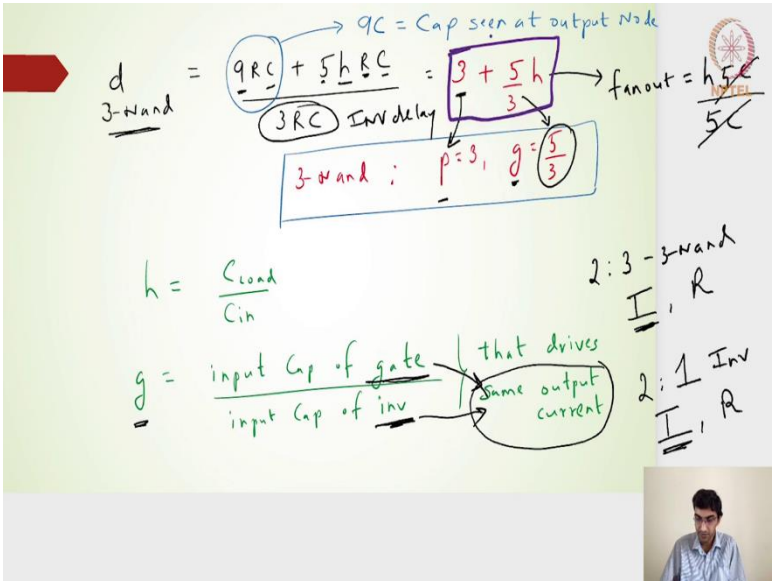
The other advantage of this linear delay method, if you consider the propagation delay falling I get 9CR and if I consider the propagation delay rising using this linear delay model, it will be nothing but this particular capacitance 9CR on the other side. If I consider this as a V_{dd} the switching resistance in the worst-case condition is still R considering one of the transistors will be on.

The rising delay using the linear delay model is also 9RC. Effectively the propagation delay rising and falling in the linear delay model, if we consider the gates to be having the equal rising resistance, the falling resistance to that of the 2:1 unit inverter we should get both the propagation delay same.

Whereas in the Elmore delay method when we had applied and calculated or estimated the delay, it was different. The rising delay was 15RC, the falling delay was 12RC. That is a major change or the difference between the Elmore delay method and then the linear delay model.

Although the Elmore delay method is much more accurate, the linear delay model is just a model just for a faster estimation of the delay. Hope you have understood this. Moving forward.

(Refer Slide Time: 09:31)



If I use the linear delay model for a 3-input NAND gate, I will have this 9RC which we had estimated in the previous slide. If I have the second stage I have h number of 3-input NAND gates, I will get,

$$d_{3-NAND} = \frac{9RC + 5hRC}{3RC} = 3 + \frac{5}{3}h$$

$$3 - NAND: p = 3, g = \frac{5}{3}$$

$$h = \frac{C_{load}}{C_{in}} = \frac{h5C}{5C} = h$$

h which reiterates that h is not the number of gates it is or it is not the number of wires that is going to the next stage. It is basically how much of the load capacitance is the ratio of the load capacitance with respect to that of the its own input capacitance.

g is the logical effort and p is the normalized parasitic factor which is 3 and 5/3 respectively. This once again I have explained the $h = \frac{C_{load}}{C_{in}}$ of its own stage. Logical effort

$$g = \frac{\text{input cap of gate}}{\text{input cap of inv}}$$

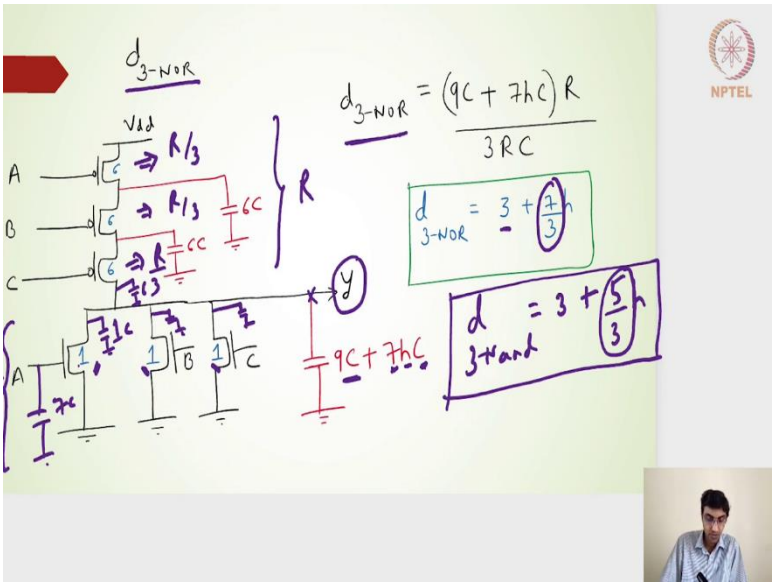
It is basically 5 /3, it is basically a ratio of the gate input capacitance and that of the inverters input capacitance. We are basically checking the driving strength of the gate with respect to that of the inverter.

Although I have put here, the ratio should be accommodated in such a way that we should get the same output current. What it means is if I am taking 2:3, 3-input NAND gate I am having the output node capacitance actually discharging at a current of I.

Whereas with respect to the inverter of 2:1, I will have this particular 2:1 inverter is also going to discharge with the same current of I. Because in both the cases we have adjusted the sizes on the NMOS and PMOS so that we will get the resistance falling and the rising resistance of R for the 3 input NAND gate as well as for the inverter.

The capacitor discharging current will be I and if that is the I value then it will have the same I for the 2:1 inverter. While we are defining the logical effort should always be defined for a gate by benchmarking the inverter size so that both of them gives the same output current. We will take a deeper look into it in the future slides in the future lectures. Hope at this point of time you have understood what is p, what is g and what is h.

(Refer Slide Time: 14:00)



Moving ahead, we had seen for 3-input NAND gate. What should be it 3-input NOR gate? For a 3 input NOR gate we will adjust the sizes, we will design the sizes of the transistor such that we will get the rising resistance of R and then the worst-case falling resistance to be R.

For a 3-input NOR gate we will have on the pull down side all the 3 transistors to be in parallel. On the pull up side we will have all the 3 transistors to be in series. If I have each of this transistors giving me R/3 somehow R/3 and R/3 then we will have the rising resistance to be R which will match with that of the 2:1 inverter rising resistance

For that reason, we will have the size to be 6 for a PMOS transistors and on the worst-case condition one of the transistors on the pull down side if it is on I will get the falling resistance to be R. That means, I will have the sizes to be nothing but 1, 1 and 1. Once we have identified the sizes what should be the linear delay model. I need to establish what should be the capacitance seen at the output node.

The capacitance is coming from this particular transistor 1, this particular drain to body capacitance is 1C. There will be a drain to body capacitance of 1C here and then there will be a drain to body capacitance or 1C here. The 3 capacitances will contribute to the output node. In addition to this particular there will be a 6C capacitance from this PMOS transistor, overall I will see a 9C capacitance.

If it is cascaded, if the output is going to the next stage, the input capacitance for the A input will be nothing but 1 + 6 = 7C. For the B, it will be 7C, for the C it will be 7C. The next stage if it is connected as one of the inputs to one of the A, B or C it will have a 7C and if I have 7 such NOR gates in the next stage I will have 7hC.

The normalized delay for the 3-input NOR gate will be nothing but,

$$d_{3-NOR} = \frac{(9C + 7hC)R}{3RC}$$

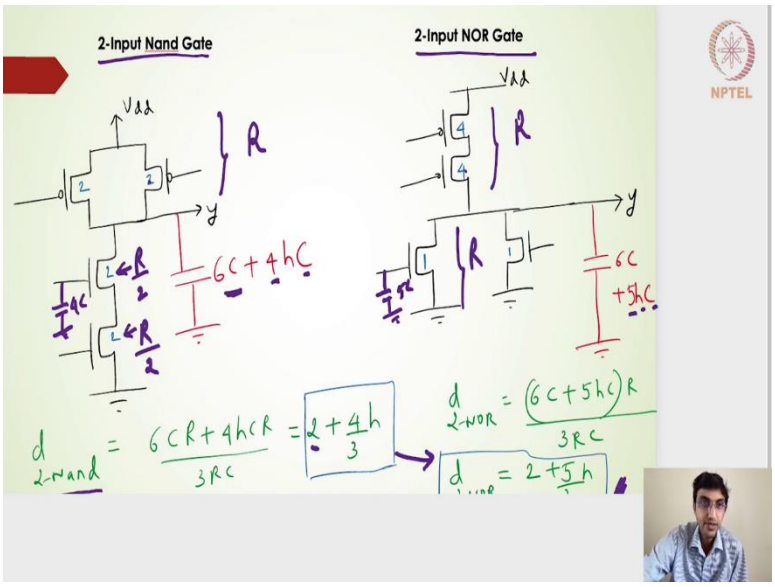
$$d_{3-NOR} = 3 + \frac{7h}{3}$$

Remember for the 3-input NAND gate we had a $3 + \frac{5h}{3}$ and then for the 3-input NOR gate it is $3 + \frac{7h}{3}$.

Just to verify that same, if I go back to the previous slide I will get this $3 + \frac{5h}{3}$ this particular value. If I go back to the present slide, we can easily say that the NAND gate is much more faster because it is $3 + \frac{5h}{3}$. This 5/3 will be a lower value than 7/3 value, the NAND is actually much much faster than that of the NOR gate.

In any kind of a designed case if we can accommodate the NAND gate instead of a NOR gate, one will go for the NAND gate because it is much much faster, the delay is actually less. Hope you have understood this.

(Refer Slide Time: 17:36)



Moving ahead. I have also drawn 2-input NAND gates and then the 2-input NOR gates. The sizes for a 2-input NAND gates will be on the PMOS side it will be 2 and 2, that will give me the worst-case condition, it will give me a rising resistance of R. On the pull down side for a 3-input NAND gate we had a size of 3, 3 and 3, that we will get R/3, R/3 and R/3.

Here it is only 2 transistors. I need to have switching resistance of R/2 coming from one transistor and R/2 coming from the another transistor and hence we should have a size of 2. The overall capacitance seen at the output node y will be nothing but 2 coming from here, 2 coming from here and then 2 coming from here, so 6C and if it is cascaded to the next stage the input next stage input will see a capacitance of 4C in both the inputs.

If it is connected to one of the inputs on in the next stage for the same 2 input NAND gate, it will be nothing but 4Ch. The overall normalized delay in a linear delay model will be

$$d_{2-NAND} = \frac{6CR + 4hCR}{3RC} = 2 + \frac{4}{3}h$$

It says that a number of inputs, if I consider the 3-input NAND gate, it was $3 + \frac{5h}{3}$ which is more than that of the 2-input NAND gates. A 2-input NAND gate is more than that of the inverter, which is nothing but $1 + 1h$

For a 2-input NOR gate and made the sizes of 4 and 4, that effectively I will get the switching resistance as R/2 + R/2 as R and on the pull down side if one of the transistors is on the NMOS side I will get the resistance of R here.

Overall, at the output node I will get the overall parasitic capacitance is nothing but 1 + 1 + 4 =6C and then plus each of these inputs is going to see 4 + 1=5C. If this output gets connected to one of the inputs of the 2-input NOR gates then I will get 5C and then h number of NOR gates in the second stage will give me 5hC.

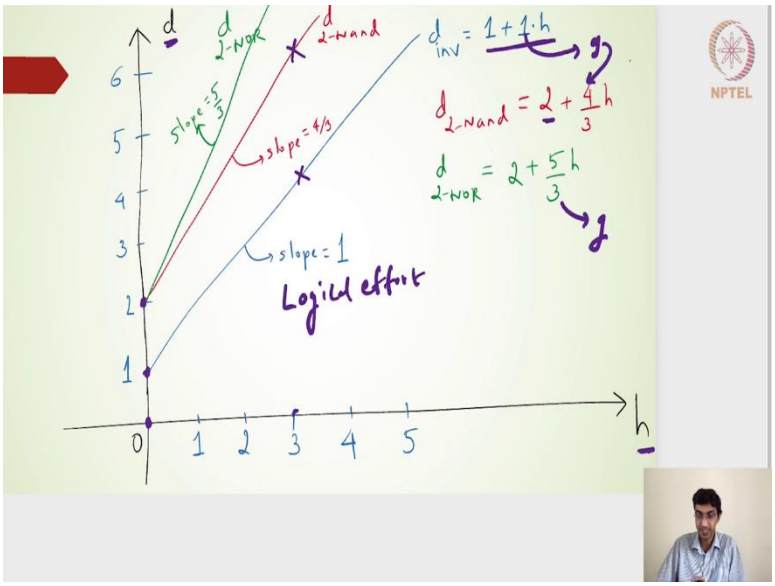
Overall, the normalized delay will be,

$$d_{2-NOR} = \frac{(6C + 5hC)R}{3RC} = 2 + \frac{5}{3}h$$

This is much lower than $3 + \frac{7}{3}h$ which was for the 3-input NOR gate, but it is slightly more than that of the 2-input NAND gate. Again 2-input NAND gates is better than the 2-input NOR gates.

Similar to that of the 3-input NAND gates is better than the 3-input NOR gates because the delay is slightly less. At this point of time what we have seen is the linear delay model for the inverter, for the 2-input NAND gates, 2-input NOR gates, 3-input NAND gates and then at 3-input NOR gates.

(Refer Slide Time: 20:42)



This is a basically a graph of the normalized delay with respect to the number of h the electrical effort or the fan out, d versus h. If h = 0, the electrical effort is 0. That means, the effort delay is 0, that means the first stage is not putting any effort to the second stage. What it means is if h is equal to 0, that means the delay is only due to its own parasitic capacitances or the parasitic factor.

In that case if I consider the inverter, the inverter has the normalized delay, linear delay model $d_{inv} = 1 + 1h$. If h = 0, it will be 1 which states that its own parasitic delay is actually 1. If I consider 2-input NAND gate it is nothing but $2 + \frac{4}{3}h$. That is 2-input NAND gate is this one, it will start from 2 because when h = 0, it is a 2, this 2 value denotes that it is due to its own parasitic capacitance there is a delay of 2.

Then I will have this particular profile. Remember that the g value here, the g value here the g value here and of course, the g value here is nothing but the slope of this particular linear delay profile. The linear delay with respect to h, if I draw the slope will be nothing but the logical effort. This slope represents the logical effort and based on the h, I should be able to estimate what is the delay.

Based on if h is equal to 3 or something I should be able to find out what should be the inverters delay, what should be the 2-input NAND gates delay, what should be the 2-input NOR gates delay, what should be the 3-input NOR gate and then 3-input NAND gate. If I

consider the 2-input NOR gate here on this particular line here, the slope is 5/3 and as the h increases I will get more and more delay.

This particular linear delay profile also confirms that the delay parameter is nothing but with respect to h value, it is a linear function. If I actually draw the Elmore delay model on this you will see that it is a non-linear function, that is the basic difference. The linear delay model if I draw the profile of delay with respect to the h value we will see that it is a linear rough profile. It is a linear function with respect to h.