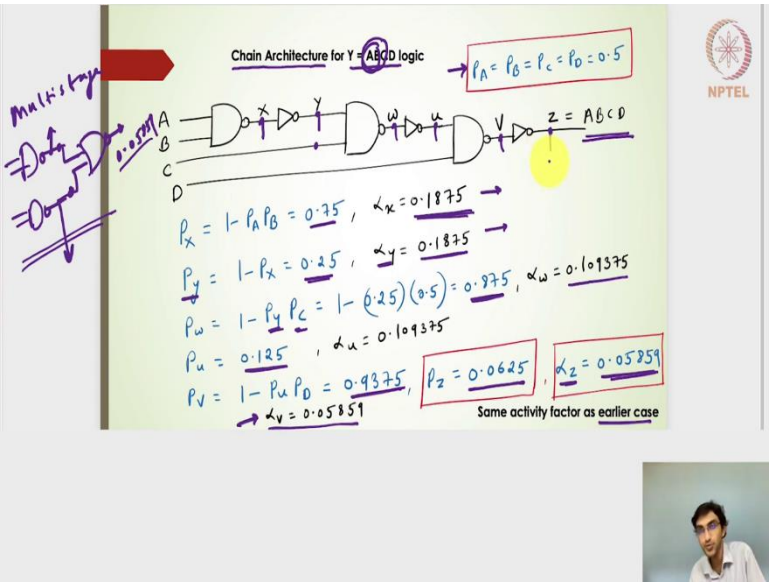
Design and Analysis of VLSI Subsystems
Dr. Madhav Rao
Department of Electronics and Communication Engineering
International Institute of Information Technology, Bangalore

Lecture - 64
Analyzing Dynamic Power

(Refer Slide Time: 00:16)



Hello students. Welcome to this lecture where we will continue on the activity factor and using that particular activity factors we will try to evaluate the overall or the average power or the dynamic power, the switching power as well as the switching energy and then the dynamic energy. The last time what we had seen is an output ABCD as a logical expression.

We had two input NAND gates and then followed both of them were connected in the next stage where in the NOR gate and then where we got the output as ABCD. We call this as a multi-stage architecture because it is of two stage circuit and we try to estimate the alphas each of this output nodes, the α here, the α here and then the α at the final output node.

In this particular architecture, we call it as a chain architecture the reason is there are multiple stages here and the final output is the same, but it forms in the form of it has been evaluated in the form of a chain architecture. If I consider ABCD here using a two input NAND gate I will get $\overline{AB}$, if I supply the input of A and B.

The output of this particular two input NAND gate can be supplied to the inverter, and then here the output will be AB. Supply to another two input NAND gate with a C input I will get $\overline{ABC}$ inverted output will give me A B C. Supplied that to the two input NAND gate with the D as another input we will get $\overline{ABCD}$ and then with the inverter we will get ABCD, this is my overall logical expression. If I consider this particular method and then this particular method the overall logical expression remains the same. But one we call it as a multistage, here we call it as a chain architecture, and you can consider this because we evaluate AB first and then ABC first and then we will evaluate the ABCD. In that sense also, it can be considered as a chain architecture. The one thing to note here is I have a lot of output nodes here whereas, in this multi-stage architecture I had only 3 output nodes.

We need to extract the capacitance at each of this nodes and then multiply or estimate the activity factor at each of these output nodes, and then multiply that activity factor with that of the capacitance and then do the summation and then multiplied by V_{dd}^2 will give me the switching energy or the dynamic energy. In this particular case, I actually have a lot of output nodes there is an x output node, there is a y output node, there is a w, there is a u output node, v output node, and z output node.

In fact, the number of output nodes where I need to estimate the α and then where I need to estimate the capacitances turns out to be 6 in this particular chain architecture. Whereas, in the multi-stage it turns out to be the number of output nodes is nothing but 3. Let us try to evaluate the activity factor and then try to estimate the overall energy term or the power term.

The probability at the point x, the x being 1 is nothing but it is an output of a 2 input NAND gate, by considering the same input probabilities of 0.5 across all the inputs. We will get,

$$P_X = 1 - P_A P_B = 0.75$$

$$\alpha_X = 0.1875$$

It is very similar to that what we had got for the multi-stage architecture at the node x. At the node y I will have the inverted output of the x input. It is nothing but,

$$P_y = 1 - P_X = 0.25$$

The activity factor will be nothing but P_y and then complement of P_y. It remains the same as that of the P_x or whatever the activity factor at the x node. So, even the y node will give me an activity factor of,

$$\alpha_y = 0.1875$$

Remember that the y node output is an output of the inverter. In the multistage architecture, we had got the output here as 0.875, the activity factor at that output node here was nothing but 0.1875. P_w which is nothing but the probability at the output of the second two input NAND gate is nothing but,

$$P_w = 1 - P_y P_c = 1 - (0.25)(0.5) = 0.875$$

$$\alpha_w = 0.109375$$

The P_u which is nothing but the inverted output of w, I will get,

$$P_u = 1 - 0.875 = 0.125$$

$$\alpha_u = 0.109375$$

The P_v is nothing but I know it is coming from the third two input NAND gate. I will have to, I mean it is nothing but a function of,

$$P_v = 1 - P_u P_D = 0.9375$$

$$\alpha_V = 0.05859$$

This is something we had got previously also this particular value and we had got this particular value of the activity factors and then finally, this P_z value which is nothing but the inverted output of the v signal which is nothing but.

$$P_Z = 0.0625$$

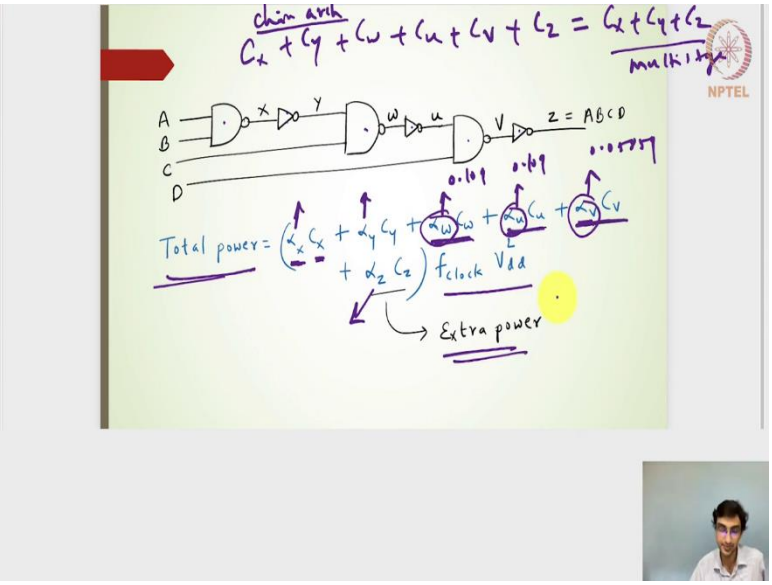
$$\alpha_Z = 0.05859$$

We achieved at the output node of z, we actually achieved the same activity factor as that of the earlier case.

If you remember I had here the activity factor of 0.1875 and here it was nothing but 0.05859, which is what we had achieved here also 0.05859. The overall activity factor achieved by this particular chain architectures remains the same as that of the multi-stage architecture and the reason is very very simple, because this activity factor are evaluated from the logical functionalities of the gates.

The logical functionalities of both the architectures being the same, we should be able to evaluate the activity factors also to be the same. In fact, we will get the same activity factor. But the difference here is, if I need to find out the overall power or the overall energy here the capacitance is here, the capacitance is here, the capacitance is here, here, and here and here, needs to be multiplied with the respective activity factors and then of course, multiplied with that of the $f_{clock}V_{dd}^2$.

(Refer Slide Time: 07:26)



This is what the total power turns out to be. I have an $f_{clock}\ V_{dd}^2 \sum \alpha C$ is, which is nothing but alpha for that particular node multiplied by the capacitance at that particular node. If I notice this my $\alpha_X = 0.1875$, $\alpha_y = 0.1875$, $\alpha_Z = 0.05859$.

But we have an extra α_w, α_u, α_v. Let me say that $\alpha_w = 0.109$, $\alpha_u = 0.109$, $\alpha_v = \alpha_Z = 0.05859$. If I even consider the widths of all this 1, 2, 3, 4, 5, 6 gates, I will specify some kind of a width. That the overall capacitance of

$$\text{Total Power} = (\alpha_X C_X + \alpha_Y C_Y + \alpha_w C_w + \alpha_u C_u + \alpha_v C_v + \alpha_Z C_Z) f_{clock} V_{dd}^2$$

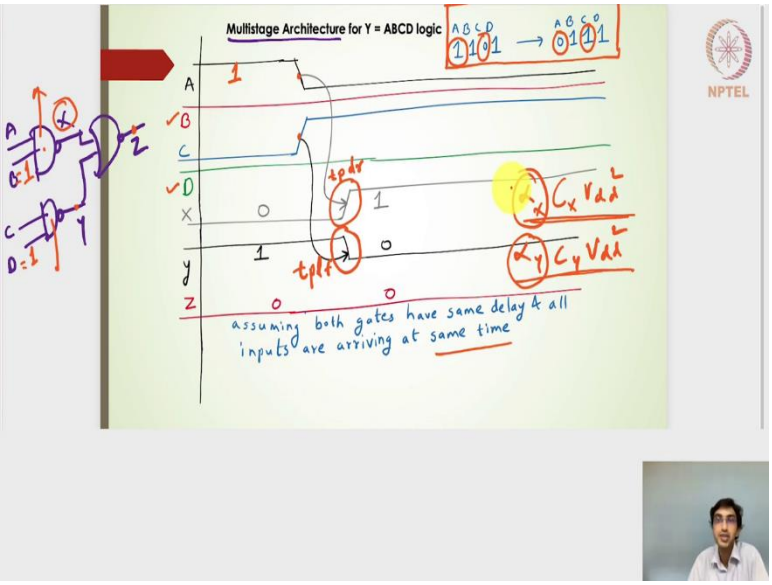
This is my multi-stage design, and let us say this is the architecture design, the chain architecture design.

If I do that the chain architecture and then the multi-stage design, even if I say that all the parasitic capacitances or the capacitances seen at the inter nodes of all this of both the circuits if it is same. Even then my alpha values there is an extra α_v, α_u and α_w which is going to give an extra power. What I have done is in this particular case $\alpha_v = 0.1875$ and $\alpha_Z = 0.1875$. We can choose one of them.

If I choose α_v with that of α_x and α_y being that there in the previous case, but still I will have α_Z, α_w and α_u multiplied by some capacitances and then the overall power is going to increase. Having more nodes or having more stages especially in the chain architecture, although it is going to give me the same logical expression, but still I might have an extra power.

I will be able to know this particular chain architecture is going to dissipate or the V_{dd} rail has to deliver more power or more energy for the chain architecture.

(Refer Slide Time: 10:15)



There is one more aspect which we tend to ignore while calculating the switching energy term or the switching power term and let us try to see that in this particular timing analysis. Let us go back to that multi-stage architecture. Again, I am going to draw that multistage

architecture. I have two input NAND gate followed by the NOR gate. I will have A B C and D and then here is my x, here is my y, and then this is z, this is my OR gate and then these two are two input NAND gates.

Let us take an input here the input is ABCD which changes from 1 1 0 to 1 to 0 1 1 1. It is basically an AND gate. The output here for ABCD at the output at this z node the final output basically should remain to 0. In a multi-stage architecture, let us try to draw this particular transitions, the input starts from 1.

If I look into this particular expression of ABCD changing from 1 1 0 1 to 0 1 1 1. It is nothing but A is actually changing from 1 to 0 and B remains the same, D remains the same, C is changing from 0 to 1, that is about it. The B and D remains the same, remains 1 throughout. My timing diagram says that B remains 1 throughout, D remains 1 throughout, A is changing from 1 to 0, and let us say that C is changing from 0 to 1 at the same time.

When the transition of A happens from 1 to 0, at the same time the C also does the transition. Let us take a very ideal case and let us try to identify what happens. Let us try to accommodate this particular input transitions into our multi-stage architecture as well as the chain architecture and then try to evaluate the final output expression.

If I have this ABCD where B and D are doing the transitions or rather B and D remains 1 and A and C are doing the transitions the output of x here, the output of x will be nothing but if B is set to 1, the output of x is nothing but an inverted output of A. The A and B NAND gate, so 1 and 1, it will be 0 throughout, till A does the transition from 1 to 0.

The moment it goes to 0, A goes to 0, I will get an x going higher and that is what it is, this is nothing but I will say propagation delay rising. With respect to the timing parameters I say that when the A goes down that is when the x is going to go high, there will be some kind of a delay. There will be a gate delay for this particular NAND gate and that will be characterized by the propagation delay rising.

Let us consider the y output. The y output is nothing but C and D, where D is set to 1 throughout, I have a D which is continuously 1. The C is actually doing the change from 0 to 1. Again y will be nothing but an inverted output of C. Whenever the C changes, there will be a change in the y. When the C was initially 0 I will get a y to be high and then when

the C does the change y is going to C, does the change rising change now y output will be low and because C and A are doing this at the changes at the same time I will also get.

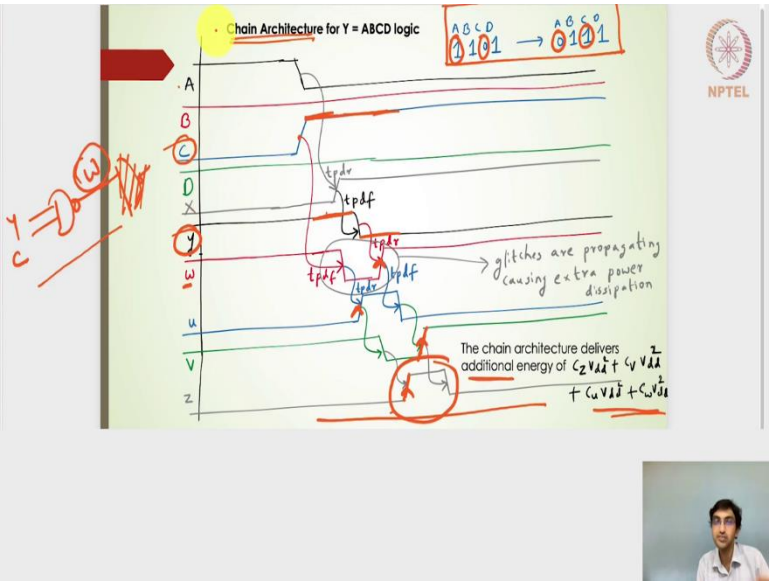
Let us say that this gate and then this gate, are actually having the same kind of a width or sized in a very similar manner then I get the same delays. For the y output I will get the falling delay with respect to the C, whenever the C starts rising I will get the y output to be falling. I will get this as nothing but the propagation delay of falling.

Finally, the z which is nothing but the NOR, the NOR of the output x, of the input x and y. The output z, if I consider x and y where 0 and 1, the output z will be nothing but a 0 here and here also it will be 1 and 0, it does the transition at the same time. The 1 and 0 I will get the output of 0. Assuming both the gates have the same delay and all the inputs are arriving at the same time, this is what we will get the z output.

If I consider the x output here, the x is doing the transition here and then y is doing the transition here and somewhere else it will do the higher transition again. This particular alpha activity factor of x accommodated with that of the capacitance at C_x, that we will get the energy term CV_{dd}^2 and $\alpha_y C_y V_{dd}^2$. This particular what do you say a legitimate transitions especially in this case and then in this particular case which is arrived because of the logical expression of this particular two input NAND gates.

Both these expressions one going high and one going low have been attained or have been achieved by this logical functionality of the two input NAND gates. Whatever is this transition, this is anyways been accommodated in our energy term because alpha y accommodates the logical functionalities. It actually depends on, alpha x depends on the probability of the inputs being 1 and 0.

(Refer Slide Time: 15:59)



Moving ahead let us take the chain architecture. We have the chain architecture that means, A and B and C and D, A and B will create x and then x inverted will be y, C and y will give me w, w's inverter will be u, u and D will give me v and then the inverted output of v will be z. Let us say I will take the same example the input doing the transitions in this case being ABCD starting from 1 1 0 to 1 1 0 1 to 0 1 1. Again, the same thing A is changing from 1 to 0 and C is changing from 0 to 1 and B and D are retaining at the logic level high, that is what I have done. The B and D are written at logic level high A is doing the transition, C is also doing the transition and just to benchmark against the multi-stage architecture what we have done is we have taken the transition at the same time. The A and C doing the transitions at the same time. Let us try to find out what is x. The x will be nothing but the inverted output it is a two input NAND gate where one of the input is high. It will be nothing but the inverted output of A with some particular delay.

Whenever A is high, I will get x to be 0 and whenever A goes low I will get x to be high. Then this particular delay is called as propagation delay rising off the first two input NAND gate. The y is nothing but the inverted output of x. The y will start with high and then whenever x goes high y will go low. I will get based on the propagation delay falling of the inverter, I will get this particular signal of y.

The w is nothing but the two input NAND gate of y and C. It is nothing but the two input NAND gate of y and C. This is kind of little bit interesting here. That is what the w is. So,

when C is low and y is high, the output should be high and that is what it does. The moment both of them in this particular case, because of the delay in y, both in this particular region when y is high and the C is also high here.

I will get the output to be low and that is what based on this particular C B going high I will get this particular delay due to this particular delay or the transition state I will get the w output going 0. This will be propagation delay falling and then C stays at high, but y goes low here. During this particular transitions, I will get 0 and 0 at the inputs of the two input NAND gate, my output will be high.

That means, that w has to go high again. The w goes high again what it implies is, w going high I will get unnecessary transition here, this circle which I have drawn here, it is an unnecessary glitch or an unnecessary transitions one going low and one going high just because of this particular propagation delay.

The t_{pdf} and then the t_{pdr}, what it means is the y which is kind of delayed due to the x, the x kind of getting delayed by the gate delays, but the y the inverted output will have some kind of a delays, and then the delay in the y will is going to give me this kind of a small glitch of going low and then coming back high.

This particular transition is a glitch. What happens is the output of w is then passed to an inverter. The inverter also shows or throws a glitch again here which is an inverted glitch of this. I will have when w is high and u will be 0 and then we will have this going low and high, in the u it will go high and then come back to low, there will be a glitch here which is kind of propagated from the w signal.

Then this particular one will finally go to the two input NAND gates, the w output or rather the u output will go to the two input NAND gates where the other input is D, where, but the D is a constant one. The output of the two input NAND gate, the 4th means x being the one and then w being the second one and then of course, the third two input NAND gates in the chain architecture we will get because D is one I will have an inverted output of u.

I will have a u, whatever is u the inverted output of u it will be high and then when it goes high and low it will go low and high. Again this particular glitch is kind of propagated to v starting from w to e to v via u. Lastly the z output which is nothing but the inverted

output of v. This particular glitch is kind of propagated from w till the z output, till the final output.

If I consider the logical transitions here if I consider the output z transitions. What I really want is the z to be staying low throughout. But here I have this particular transition. What the transition indicates is the capacitance at the z output is going to get charged, capacitance at the v output is going to get charged, capacitance at the u output is going to get charged, the capacitance at w output is going to get charged.

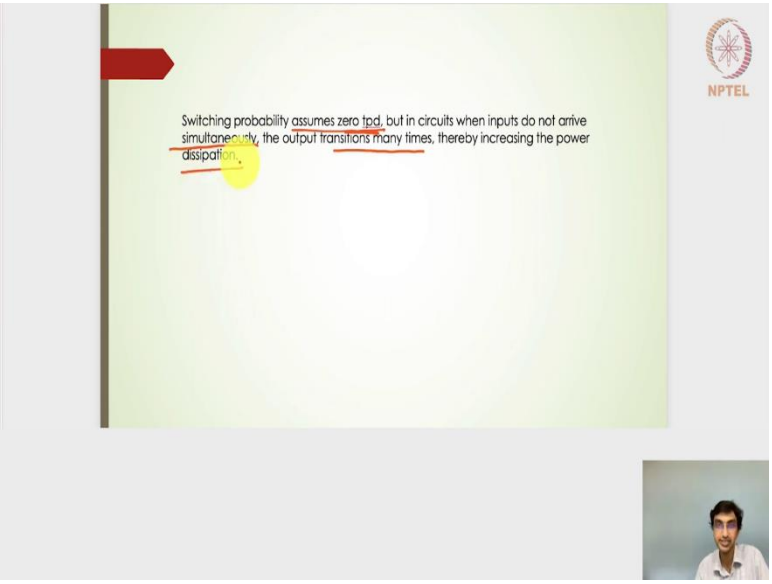
The glitches add an additional transitions which is not accommodated or not incorporated in our overall estimation of the activity factor. Remember that this particular glitches are coming not because of the logical functionalities of the gates, but because of the delays. If we do not actually accommodate the delays here, I will actually not get this particular glitches at all.

If I consider the delays are 0, I will get the transitions whenever the C is doing the transition, that is when the x is going to do the transition, that is when the y is going to do the transition that is when the w is going to do the transitions and so on. In that case I will not have this glitch at all. What they happens is because of this delays I will get an unnecessarily transition at the output and then it has to go back to 0. An unnecessary transition of the output node from 0 to 1 will cause an additional energy or the power to be delivered by the V_{dd} rail.

The chain architecture delivers an additional energy of CV_{dd}^2, $C_Z V_{dd}^2$, $C_V V_{dd}^2$ and then $C_u V_{dd}^2$ and $C_w V_{dd}^2$, which is not accommodated in our α calculations for the energy term, these are additional ones.

That is something one has to take care and that is why having more number of output nodes or I have especially in the chain architecture, it is not a feasible you know it is not an energy aware design. If I really want to have the power benefits or the energy savings then actually we should go towards the multi-stage architecture design.

(Refer Slide Time: 23:35)



The switching probability assumes the zero t_{pd} because it always has been defined by the logical functionalities of the gates and it does not accommodate the delays at all. But in circuits when the inputs do not arrive simultaneously. The output transitions will be many times, we will see more frequency of the output transitions and thereby increasing the power dissipation or the energy delivered by the V_{dd}.