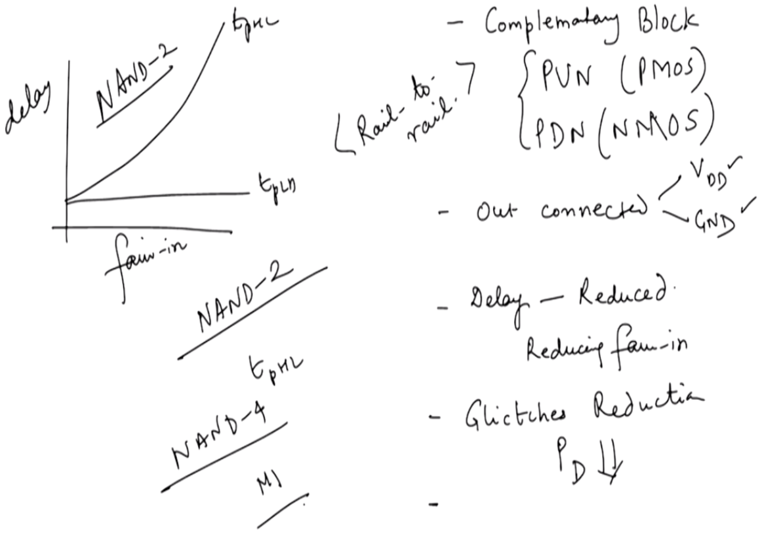
CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology – Roorkee

Module No # 03
Lecture No # 15
Combinational Logic Design – IV

Welcome once again to the NPTEL online certification course on CMOS digital VLSI design and we start the fourth module of combinational logical design in today's lecture. Before I move forward let me therefore recapitulate what we did till now as far as combinational logical block is concerned.

(Refer Slide Time: 00:47)



We looked into the fact that any combinational logical block can be made complimentary using complimentary blocks right and therefore you will have pull up networks which is basically made up of PMOS's and you also have pull down networks which is primarily made up of NMOS's right we have already learnt this point and the reason was the pull down will give you ensure a full 0 swing and pull up will ensure using PMOS's will ensure a full VDD swing.

So you get rail to rail swing in CMOS so I get what important fact is that it is a rail to rail swing right it is a very important property for complementary CMOS not only that at every point of time your output mode is always connected to output node is connected to either VDD right or to ground right that is low impedance node right because it is a voltage source and it is a ground.

And therefore your noise margins relatively higher we also learnt that various techniques or ways in reducing the speed or increase in speed or reducing the delay we learnt it yesterday at the end of the pervious lecture was that we learnt that how we can reduce the delay right. How did you reduce the delay we reduced it by reducing the fan in right fan in if you reduce the fan in you will reduce (()) (02:19).

We also saw that if you have a logic block your and you are using for NAND 2 logic right if you are using a NAND 2 logic right then since your NMOS's are always in series to each other therefore if you increase the fan in there will be an exponential increase in the value of tpHL and therefore there will be almost quadratic raise in the value of tpHL right but tpLH will almost remains constant independent of this one tpLH if we plot fanning fan in versus delay.

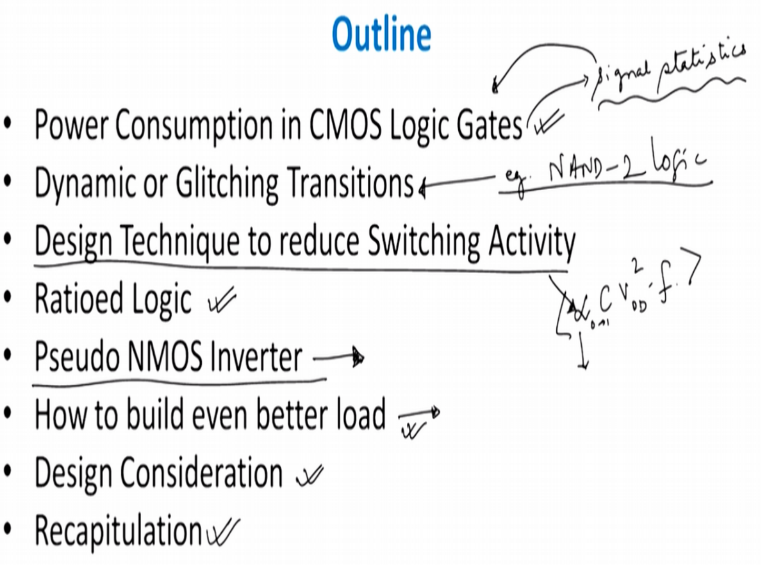
So this are the few important (()) (03:01) then we also then the reason being that you only have to switch one design or one PMOS to make output go to VDD in case of a NAND2 logic so this was the NAND2 logic right. Now we also saw in the previous discussion that by making my inputs as symmetric as possible I will reduce the number of glitches and therefore my glitches there will be a reduction and as a result there will be a reduction number power dissipation will reduce because of glitching.

How do you do that? You ensure that the signals will always reach at a particular gate exactly the same instant of time right. So that you do not have to wait for the signal to come right as the result what will happen is that you do not allow this other signal to fall below a critical limit. We also saw in the previous term that if you want to make fast systems or system which are very fast try to keep or if you want to reduce the delay drastically try to reduce or increase the W / L ratio of the that transistor which is furthest away from the output right.

So as I yesterday when we are looking at NAND4 logic we saw that in NAND 4 logic the transistor which was M1 which is furthest away from output closes to the ground that if you manipulate that if you increase the size you get a very large increase in the value of tpHL in an NAND 4 logic right. So this is one thing the second point related to this only is that if you are able to also sustain the fact that the critical signal how did you define the critical signal that signal which is finally coming to the logic block and changing the out is defined as the critical signal.

If you able to keep the critical signal closes to the output you get the fastest delay available you get a reduced delay right. So these we have already learnt in our previous section and you have also understood how these work or how these primarily so work. So we went for transistor sizing we went for input reordering we also went for glitching reduction making it symmetrical in nature and we also looked into the fact that can we have a un-ratioed logic for the purpose of reduced power dissipation with this let me start today's topic. And this today's topic is basically combinational logic design we still carrying on the combinational logic design.

(Refer Slide Time: 05:29)



Outline

- Power Consumption in CMOS Logic Gates ✓ → signal statistics
- Dynamic or Glitching Transitions ← eg. NAND-2 logic
- Design Technique to reduce Switching Activity
- Ratioed Logic ✓
- Pseudo NMOS Inverter →
- How to build even better load ✓
- Design Consideration ✓
- Recapitulation ✓

$\alpha_{0\to1} C V_{DD}^2 \cdot f$

And we will be looking at we also dealt in detail but we will be looking at the power consumption in CMOS logic gates and what factors does it depends and I will also try to show to you not only depends on the sizing of the transistors but it also depends upon the signal probabilities how the signal statistics. So we will show in this case that how signal statistics right they influence the power which means that do you have large zeros do you have large ones or you have equal probabilities of zeros and ones coming in to the stream so and hence so forth.

On that factor also you will actual have your power dissipation for this logic gate we will explain this directly (()) (06:14) thus we will look into what is known as dynamic transitions or glitchings rights and we will take an example of a NAND 2 logic. So this example which will be taking in this case will be basically a NAND 2 logic we will take a NAND 2 logic and explain to you what is the meaning of glitching and how does the glitching influence the power dissipation of a standard CMOS process.

Now once you have understood this point I think it is very important if you remember my starting whereas (()) (06:45) power dissipation we discuss at alpha C times VDD square multiplied by F where alpha was defined as the switching activities right alpha 0 to 1 right. So what we will be doing in the third part of the talk is we will looking into ways and means of using this alpha right.

So how can I reduce this alpha to a lower value without compromising the functionality of the chip right. So that is very important that though you are told to reduce alpha you have always ensure that the overall functionality of the chip is not getting damaged or it is not getting reduced or it is not getting changed right and that is pretty critical as far as understanding is concerned then we will look at ratioed logic right what is the meaning of ratioed logic?

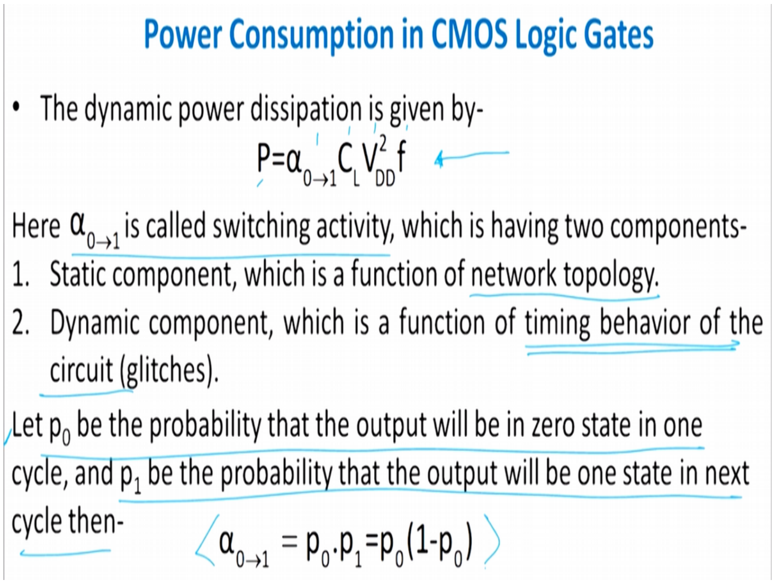
As I discussed with you previous term there are two types of logic available in standard CMOS technology one is the NAND ratioed logic one is ratioed logic. Ratioed logic primarily means that where W / L ratio also influences the functionality right so in the W / L ratio is the aspect ratio of either the pull up or pull down transistor effects the functionality of your chip then we define that to be as ratioed logic, if it does not we define that to be a NAND ratioed logic right.

Functionality please understand I am not saying it is tpHL, tpLH it is power dissipation so on and hence so forth. So NAND gate as to work as a NAND gate only right then only it define that to be as a functionality we will look at the Pseudo NMOS inverter once example of a Pseudo NMOS inverter right what is this Pseudo NMOS where is simple and straight forward till now we were doing till now we were actually looking into a PMOS and NMOS so if I replace the PMOS which is the pull up network by a resistor right.

It is still work as a inverter we will see that but there will be certain disadvantages of those situations and we will try to study those when we discuss with you Pseudo NMOS inverter right we will also looking into fact that if we have understood what is ratioed logic and unratioed logic so can we have an alternate form of logic by which I can get not only the output but I can get also the inverts of output right.

So we will looking that in this case right and look at the design consideration then recapitulate finally at the end of the show at the end of the whole discussion on how it works out right. So this is general topic or general idea which we will be looking into let us come to the first section of the talk and CMOS based digital design and this is basic as combinational logic design module 4 or section 4.

(Refer Slide Time: 09:26)



Power Consumption in CMOS Logic Gates

- The dynamic power dissipation is given by-

$$P=\alpha_{0\rightarrow 1}C_L V_{DD}^2 f$$

Here $\alpha_{0\rightarrow 1}$ is called switching activity, which is having two components-

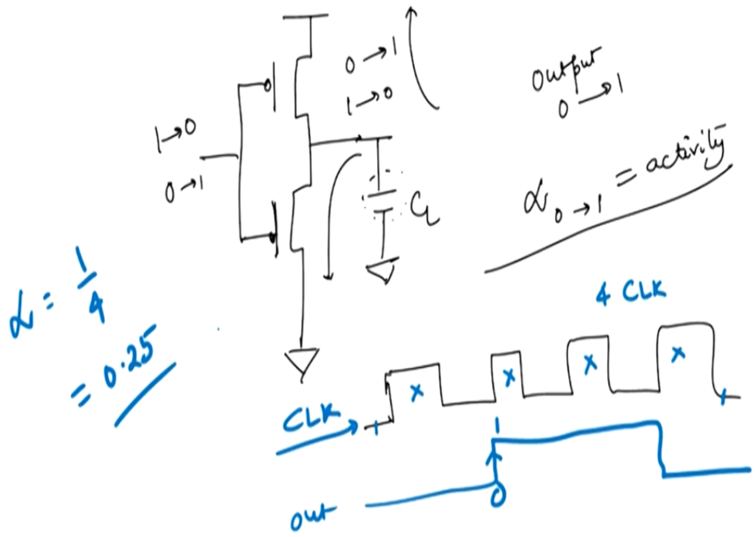
1. Static component, which is a function of network topology.
2. Dynamic component, which is a function of timing behavior of the circuit (glitches).

Let p_0 be the probability that the output will be in zero state in one cycle, and p_1 be the probability that the output will be one state in next cycle then-

$$\alpha_{0\rightarrow 1} = p_0 . p_1 = p_0(1-p_0)$$

In this I have already discussed with you in this earlier days that P was equal to then this is given by this that you have alpha 0 to 1 right CL VDD square in to F is P from where did you get it let me just recapitulate or make your memory I will refresh it.

(Refer Slide Time: 09:44)



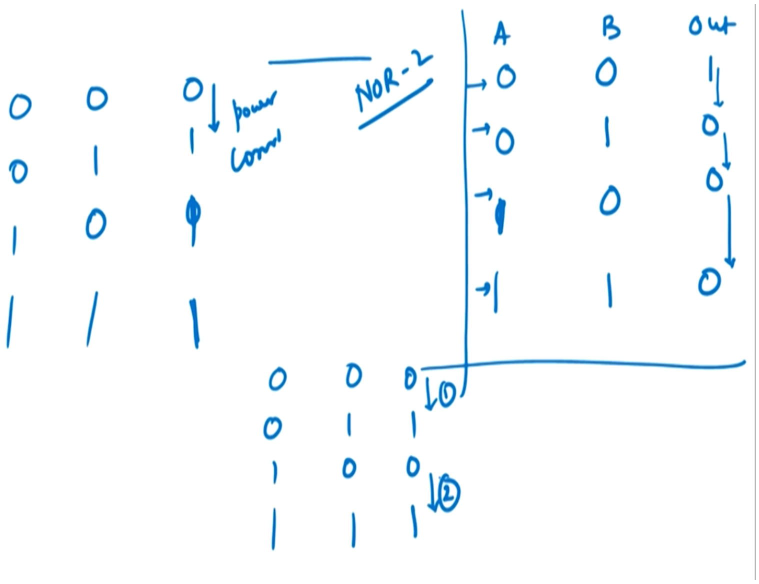
So if you remember that if I have a PMOS here and NMOS here right sorry this is NMOS and I join them together and I give a signal input going from 0 to 1 and the output going from 1 to 0 right, So going 1 to 0 means primarily it was high and then it goes to low so if it is 1 to 0 in the output input will be 0 to 1 here and therefore the output will be going from low to high. Now please understand that this is CL so capacitor will be charged in this charge during the transition when the output is going from 0 to 1 right when the output is going from 0 to 1 you are actually consuming power from the VDD rail because you are actually extracting power from the VDD rail and storing that in form of charge at CL.

So therefore alpha 0 to 1 is defined as the activity factor here right I will explain to I have already explained to you but just to refresh the memories how do you define activity factor. See for example if you give an input signal which is something like this maybe a clock give to the input side and therefore clock signals available to you if this is the clock which you are giving right it is the clock right the if this is my input which tells and this is my output right and these are four cycles 1, 2, 3 and 4 clock cycles right.

And if you see there are let u suppose 0 to 1 transition in the output side let us suppose so for every four clock cycle I get 1, 0 to 1 transition in the output we defined alpha to be equals to 1 / 4 which is equals to 0.25 right and this we have already explained to you in a previous modules and suppose it is clear to you that this will be the value of alpha with this value of alpha therefore so the idea here is that if you want to reduce the value of alpha make it as small as possible so that this CL times VDD square is reduced drastically to an low value.

And this alpha 0 to 1 activity factor are switching factor as I discussed with you and it is two component one it is known as static component which is topology of the output topology and this is dynamic component which is basically the timing behavior of the circuits So I will explain to you what do you mean by network component.

(Refer Slide Time: 12:17)



See for example if you take NOR 2 logic let us suppose I take a NAND 2 logic then NAND what was the if you have got A B and if you have got out here for let us suppose if we take a NOR2 you take a NOR 2 one for 0, 0 output will be 1 and if it is 0, 1 output will be 0 and then 1, 0 will be giving you 0 and 0 right.

So I get this as my truth table for NOR 2 logic right now you see here the output actually goes from 1 to 0 right and then 0 to 0 then 0 to 0 right. Let us suppose this is sequence of the input bit into that NAND gate this is the sequence this is the sequence of the output gate there are no power consuming cycle such in the output side whereas if this would have been 0, 0, 0, 0 say 1 and then the 1, 1, 0, 1 and this is 1, 0 say 1 something like this (()) (13:16) then what I say at least one output cycle as got a transition from 0 to 1 .

So this is basically a power consuming cycle right similarly let me I do some something like this I have got 0, 0 right 0, 1, 1, 0, 1, 0, 1 I get 0 to 1 I get 1 to 0 again I get 0 to 1. So depending upon the time of gate (()) (13:42) two power consuming two power consuming output available to me this is known as static part of the alpha. So how does your class truth table behave how many 0 to 1 transition as it take will be defined by the static behavior of the alpha will define the static behavior of the switching activities which is alpha.

So what is the dynamic component which is there it will depend on the this is do in detail and the subsequent because see you cannot compromise on the logic block. For example if you have a NAND 2 logic there is no way you can remove a NAND to logic we can make something else right. So the static components you cannot change too much but the dynamic component surely change because you can actually change how the alpha varies or how does the alpha vary we will see that as you move along right.

Now let us suppose you it is therefore if P0 is the probability look at this point if P0 is the probability that the output will be 0 state in one cycle then P1 is the probability the output will be 1 in the next cycle then can be defined this right I think it is very clear to you why because see as I discussed with you any.

(Refer Slide Time: 14:58)

$$0 \longrightarrow 1$$

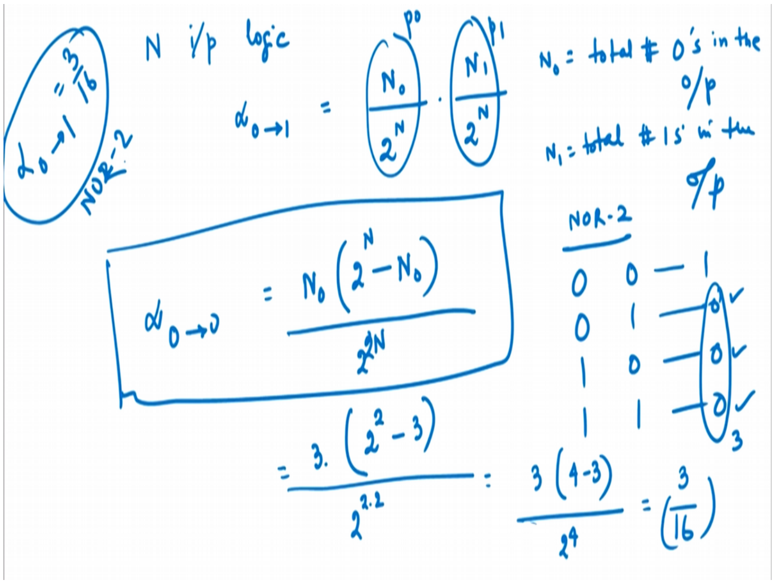
$$p_{0\to1} = p_0 \cdot p_1 \quad \text{(product)}$$

$$\boxed{p_{0\to1} = \underline{p_0(1-p_0)}}$$

You will only get a transition power consuming cycle when you have 0 to 1 cycle which means that the probability of having 0 at 1 point multiplied by probability of having 1 in the next cycle this will the probability of having 0 to 1 transition fine. So that is the basic fundamental principle that it will be product of P0 into P1 why product and why not addition very well-known right so pleas think about it why it is like that why is it a product not a addition of two probabilities it is basic mathematics to with independent your entities and non-independent entities.

Now therefore if you go back sorry therefore if you go back here I can write down P0 into 1 – P0 because P1 the probability of having 1 at a particular output will be nothing but 1 – P0 at the so this is probability of P0 to 1 right this is the probability which you get. And this basically your alpha value which you see in front of you so in a much more generalized form.

(Refer Slide Time: 16:03)



If let us suppose I have got an N input logic then we define alpha 0 to 1 as equals to N0 divided by 2 to the power N multiplied by N1divided by 2 to the power N what is N0 and N1? N0 is the total number of zeros in the output right and N1 is the total number of ones in the output right. So N0 is the total number of zeros in the output and N1 is the total number of ones if you divide by 2 to the power N total you will get the probability of this appearing so this is basically.

So this whole thing is basically P0 and this basically is nothing but P1 right fine I suppose it is clear so now if you take N0 common in the output side I get 2 to the power N – N0 / 2 to the power 2N. This is the (()) (17:12) of the alpha 0 to 1 for any 3 bit or N bit database right. So let us let me show you an example what I mean to say let us suppose I have a NAND logic there let

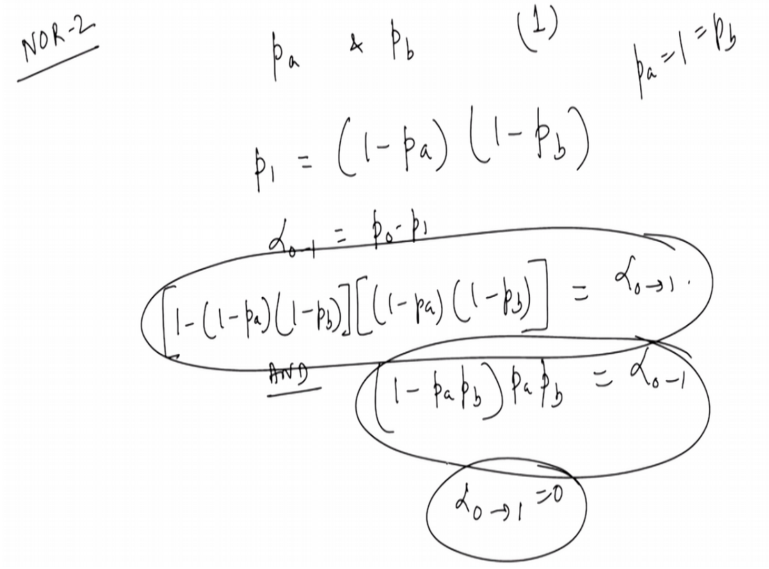
(()) (17:24) are NOR logic NOR 2 logic because it is easy to handle I get 0, 0, 1, 1 and 0, 1, 0, 1 I get this to be equals to 1, 0, 0, 0.

So what is N0 for this case number of entries 1, 2, 3 so there are 3 zeros in the output so what i do I write 3 zeros here multiplied by 2 to the power N how many number of bits are there this two bits so two bits are 2 to the power 2 – N0 how many number of zeros again 3. So I get 3 here divided by 2 to the power 2 in to 2 so I got again write down is 3 multiplied by 4 – 3 divided by 2 to the power 4 and therefore this come out to be 3 4 into -3 is 1 so I get 3 / 16 fine.

So alpha 0 0 to 1 = 3 / 16 for a NOR2 logic right similarly if your told to find for NAND 2 logic you should be able to do that in NAND 2 logic how will you do it NAND 2 logic it will be something like this that in NAND 2 logic if you have 0 output will be 1 here it will be 0, 1 output will be 0, 1 here it will be 1 here it will be 0 here right. So number of zeros here is basically 1 how many number of 2 to the power N = 2 – N0 is 1, 1 divided by 2 to the power 4 so I get 2 to the power 4 is 4 – 1, 3 / 16.

So alpha 0 to 1 = C / 16 for NAND2 logic right NAND2 logic I give it has exercise to you to please find out the values of alpha 0 to 1 for XOR logic 2 may be logic XOR logic right. And please try to find out how an XOR logic will the value of alpha will be in this case. So this is static component part of the whole design as you can see therefore what we have discussed from all these things is that something like this as the condition.

(Refer Slide Time: 19:37)



NOR-2

p_a & p_b (1)

$p_a = 1 = p_b$

$$p_1 = (1-p_a)(1-p_b)$$

$$\alpha_{0-1} = p_0 \cdot p_1$$

$$[1-(1-p_a)(1-p_b)][(1-p_a)(1-p_b)] = \alpha_{0\to 1}.$$

AND

$$(1-p_a p_b) p_a p_b = \alpha_{0-1}$$

$$\alpha_{0\to 1} = 0$$

Let me also explain to you that let us suppose that for NOR 2 logic right I say that Pa and Pb are the probability but the entities of a and b are 1 right. So I define Pa and Pb as what as the probabilities that the entities of the output at input a and b are 1 right. Probability having 1 in the output side will be 1 – Pa multiplied by 1 – Pa multiplied by 1 – Pb because it is a non-input right.

So if you solve this I get alpha 0 to 1 will therefore will look like P0 into P1 so if you break it then I get 1 – 1 – Pa right 1 – Pb if you close it down right I get 1 – Pa and 1 – Pb right. So this is alpha 0 to 1 in a more generalized fashion available to you right. Similarly for this is for NOR 2 for a AND gate I am not solving here it is 1 – Pa Pb into Pa Pb this is alpha 0 to 1 for a AND gate.

(Refer Slide Time: 20:53)

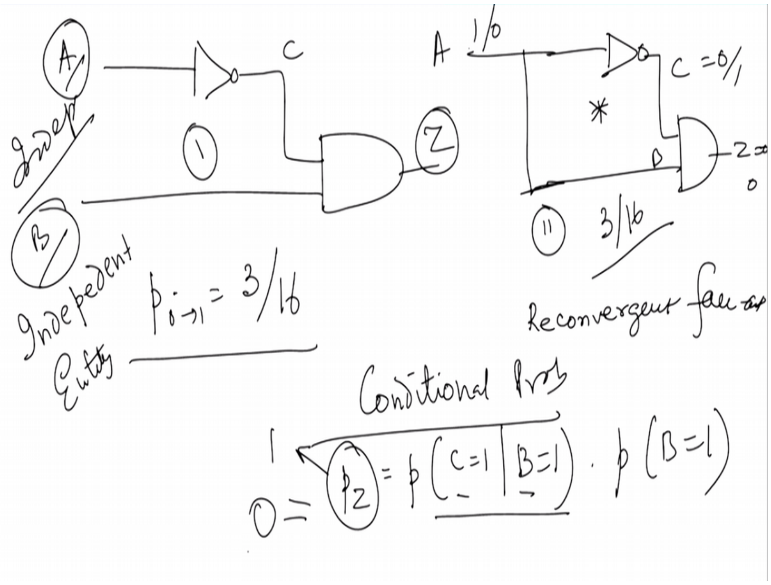
$$(1-p_A)(1-p_B)\left[1-(1-p_A)(1-p_B)\right]$$

For a OR gate it is something like this 1- PA into 1 – PB right and then you have got 1 – 1 – PA right and then 1 – PB and consider similar so on and hence so forth right that is what you will get in general overall values which you will get. Now quiet interestingly therefore if you look at these equation of these fundamental equations here which you see in front of you this one as well as for this one you will see that the transitions are therefore that depending upon the probability of having one on the input side right.

So as you can see here if the probability of Pa and Pb are both equals to 1 right Pb = 1 so what I will get from so if you put 1 here we will get this will be 0 actually. So alpha 0 to 1 will be always = 0. The reason being you will never because if Pa and Pb are both 1 right in the AND gate output will always be 1. So you will always get a 1 to 0 rather 0 to 1 transition so therefore your alpha 0 to 1 will be always = 0 in such a criteria fine.

So these things you should be careful and be aware of what these ideas are and these concepts are? With this basic concepts of concepts of logical understanding and logical idea let me again explain to you one more important thing before we move forward and that is basically.

(Refer Slide Time: 22:28)



Let us suppose I have a logic which is something like this that I have a A right and I have a AND gate and this is basically your A, B when C and Z right. Then if you do P is 0 to 1 in this case you can find it yourself it will be 3 / 16. So probability is going from 0 to 1 the output side 3 / 16 but if I do something like this that I have got this and then this right and this is something like this then also if you mathematically find out this will be still coming 3 / 16.

But in reality very well know this will be 3 / 16 the reason being these A and B I am independent in this case it was assumed to be independent entities whereas A and B I am not independent entities in second case. In first case input A and input B are defined as independent entities right entity which means that the uncorrelated signals right and therefore they are totally independent of each other whereas in the second case if you get A = 1 I will always get C = 0 and therefore by Z will be always = 0 right.

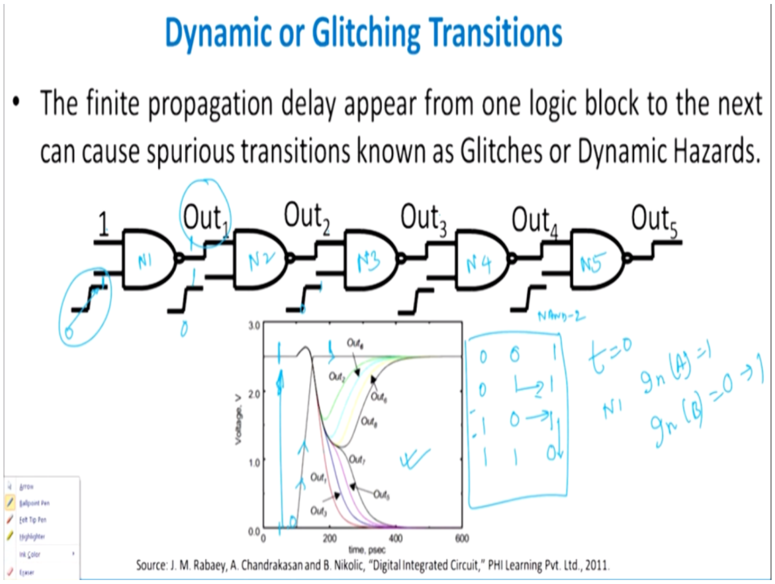
In fact if you get 0 here again this is will be 1 here and therefore you will get always 0 here so Z will be always 0 independent of my value of A and B right. This is also known as re-convergent fan out so most of you sequential logics where you have feedback path or logics which is shown here as number 2 where you have got re-convergent fan out and when the two signals are uncorrelated then this formula does not hold good really in a general sense.

So whenever the two signals are uncorrelated and statistically uncorrelated with respect to each other your this formulation is application and you get adequate this thing. Now the therefore to remove this concept people founded a new method which is basically known as conditional probability what is the conditional probability?

Conditional probability I will write down for you I said Pz have any P C = 1, B = 1 into P B = 1 which means that only when B and C when these two are equal to 1 then only you will get probabilities of Z to be equals to 1 for all of the cases you will never get the probability of Z = 1. So therefore this circuitry since C and B can never be equal logic states therefore PZ will always be equals to 0. So the probability of so the output will always lash to 0 state independent of the input state of it available to you am I clear?

So what we were understood is therefore if you have a switching network available to you these switching networks if it is a re-converging fan outs or a sequential vary of feedback you apply conditional probability principles if you have non-re-convergent and straight line path as shown in 1 then you apply the formation which we have just now discussed. With this knowledge let me move forward and show to you what is known as dynamic transition are glitching which effects your alpha.

(Refer Slide Time: 25:59)



Dynamic or Glitching Transitions

- The finite propagation delay appear from one logic block to the next can cause spurious transitions known as Glitches or Dynamic Hazards.

Source: J. M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuit," PHI Learning Pvt. Ltd., 2011.

If you look here what we have done is these are all NAND logic here so this is NAND 1 NAND so these are all NAND here right so NAND if you know very well that they will have a particular to stable available to you and I am giving an input which is going from 0 to 1 right at every part I am giving 0 to 1 and so on and hence so forth but the output 1 is coming out by virtue of the output coming from the first NAND bloc.

So let me name it n1 let me name it n2 let me name it n3 and so on and hence so forth right then 5 I have got 5 blocks and there are output 1 out 2 out 3 out 4 and out 5 which is the output of respectively of n5 independent NAND gates and at each independent NAND gate I will give 0 to 1 at the same instant of time. So at t = 0 right I have given 1 input to n1 so n1 as got 1 input as say A = 1 and input B was actually initially was 0 and going to 1 right.

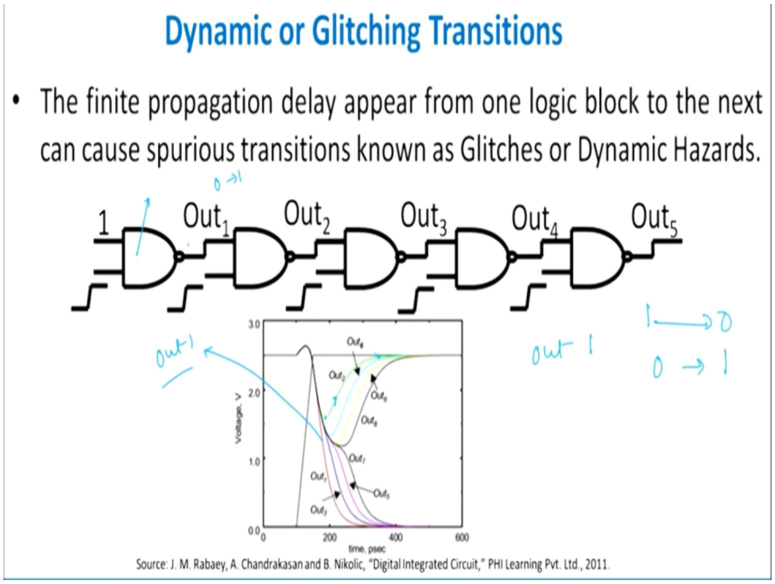
So you see let us see what happens when it was it shows if you remember your original truth table original truth table for NAND gate I get 0, 0, 1, 1, 0, 1, 0, 1 for 0, 0 I will get an output of an this is NAND 2 logic for 0, 0 I will get 1 right I will get 1 here and I will get a 0 here right. So I will get why because 0 and 1 will output will be 0 not at that will be 1 and therefore I get this is the truth table right with this truth table let us see it is graph which you see in front of you.

Now when it was initially one of the last one so get so maybe let us suppose this one and the one output one input was going from 0 to 1 right which means that let us suppose I take let us suppose I had one of the inputs as 1 and the output was initially 0 so when it was initially 0 it will be initially 1 and that is the reason the output was your this output was holding as 1 here you see here.

So whenever the input whenever my one of the input was 0 for all the outputs it was holding 1 why because by this logic diagram any of the inputs is 0 output will be 1 and that you see here and here. So I had 0 here and I have a corresponding 1 here am I clear now what I do I give a 0 to 1 transition in input side. So the input does what input goes from 0 to 1 so this is my input which you see and this is the input goes from 0 to 1 here.

As it goes from 0 to 1 right out 1 will go from 1 to 0 right I hope this is clear out 1 will go from 1 to 0 why 1 to 0 because initially it was it 1 now it is going from 0 to 1 so I will have my output going from 1 to 0 so that is the reason out 1 which you see the red curve here so you see the red curve here.

(Refer Slide Time: 28:59)



Dynamic or Glitching Transitions

- The finite propagation delay appear from one logic block to the next can cause spurious transitions known as Glitches or Dynamic Hazards.

Source: J. M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuit," PHI Learning Pvt. Ltd., 2011.

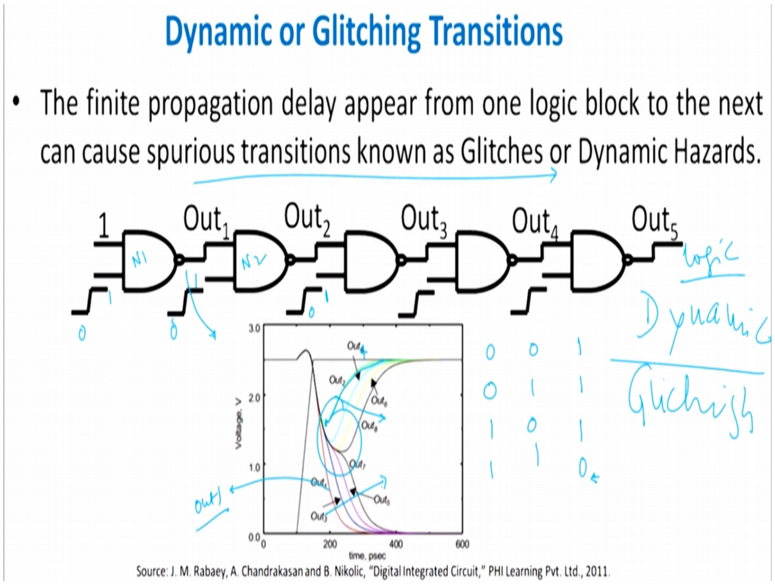
This is the red curve this is out 1 let me just reframe you out 1 goes from 0 to 1 right and it falls very fast towards the 0 mark or 0 value voltage output. As it goes so that is the out goes from **0 to** 0 to 8 it was initially 0 now it is going towards 1 when now when this both where now if both go to 1 the output of output 2 should actually latch to high value do you got the point?

So what was happening was this output out 1 as gone from 1 to 0 right at the same instance of time or approximately the same instant of time another input goes from o to 1 right. So 1, 0, 0, 1 which means that they will be a finite duration of time till which out 2 will start to fall because it is why out 2 start to because it is sees different same type of inputs at the both end but beyond a particular point it sees again a different values of voltage and therefore it is again starts to raise up are you able to get the picture this is known as glitch right.

First of all why does it fall it starts to fall because it takes a finite duration of time because these NAND gate will have a finite propagation delay agreed. So as a result since this is our finite propagation delay this out 1 will take finite amount of time for the output to go from high to low and the another input is going from low to high so the this NAND 2 will see such that both are opposite side both of same side in some point of time both are 0 and it will try to go down as it was in the previous case right.

Because so what was happening I have 0 it is this is also 0 so 0, 0 it will look as it will try to go down here right and that is what I was saying it will go down. But as it comes somewhere here this out goes from 0 to 1 out 1 goes from out 1 so this out here goes from 0 to 1 to 0 here so I will go from goes to 0 this again raises to value of the higher side right.

(Refer Slide Time: 31:26)



Dynamic or Glitching Transitions

- The finite propagation delay appear from one logic block to the next can cause spurious transitions known as Glitches or Dynamic Hazards.

Source: J. M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuit," PHI Learning Pvt. Ltd., 2011.

So I will explain I will try to explain once again the whole issue once again from the logical block concept and I will be able to give you an idea about what the glitches is all about. So NAND 2 logic is it will be 0, 0 this will be the idea so if you look at first case NAND 1 and then NAND 2 you will appreciate that this is 1 and this is going from 0 to 1 so out 1 will do what out

1 will actually start to fall down why out 1 will fall down because out 1 will goes to 0 right out 1 will start to fall down.

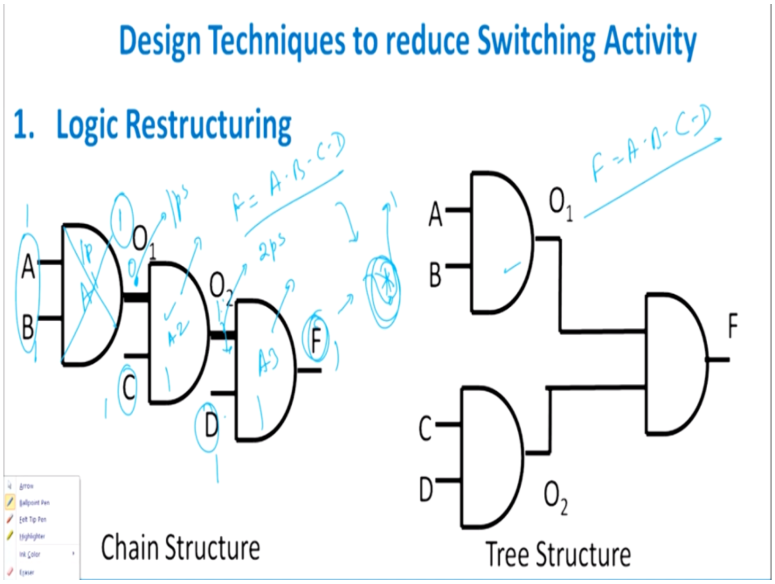
And this what happening the red curve shows you the out 1 then it starts to fall down but out 2 sorry out 1 is first so out 2 does what out 2 also 0 to 1 transition taking place. So out 2 how does it behave? It first sees that it first sees that both are equal right both are 1 so it is starts to fall down right but as it goes till this much point the other value starts to become 0 to 1 so it starts to increase. So it rather than going to 0 it goes to higher value the same concept you go and apply for higher number of NAND gates right.

And as results you will see out 4 so this is out 6 this is out 4 actually this is out 6 out 8 so all even outputs will first drop down to a minimum and then rise to highest value and all your odd bits or all outputs will go down of 0 this is known as glitching and this is known as dynamic glitching. Why this dynamic glitching? Because I really should not get this but because of finite delay of these NAND gates I start to get this glitches which results in a larger power dissipation in our case.

So the power dissipation will be much larger in this case as compared with this compared to so how what do you do it so therefore dynamic transitions or dynamic glitching will always available to you if you have large chain of such logics so try to reduce the chain and make it more symmetric in nature. When you make it more symmetric you allow a signals to be exactly the same time coming right and as a result you will have no glitching available in the outputs side.

As you can see the more I move towards the right the delay is also becoming more and more so it is more shifting towards right also both high to low and low to high right and that is understandable so it will be something like that only right.

(Refer Slide Time: 33:57)



Design Techniques to reduce Switching Activity

1. Logic Restructuring

Chain Structure

Tree Structure

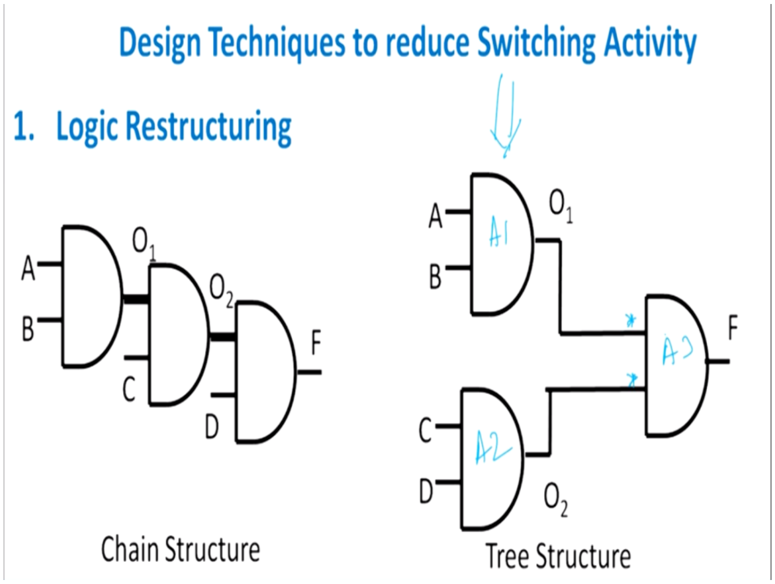
Now so what we what people have done over the years was that they did a logic restructuring that means try to make the structure as symmetric as possible right and the reason is something like this you can see here so if you look at A and B right and you see and D 4 independent signals and this is the logic which is getting realize so F will be A.B.C.D of course at here also F will be A.B.C.D.

So your Boolean expression does not show a change which means that both real as the same logic but you see what is the difference between this and this. If you look very closely here if this out input of second AND gate right this is A1, A2 and A3 so the input in the second AND gate will only appear after a delay of some value suppose this is 1 Pico second this is 1 Pico second this is 1 Pico second this is 1 Pico second.

So output at this end will only appear after 2 Pico seconds here it will only appear after 1 Pico seconds. Till that much time whatever will the value of C and D right actually your A2 and A3 will never be able to evaluate right. And as a result what will happen to the output I will tell what will happen so suppose both were equals to 1 and C was 1 D is also equals to 1.

So if both are equals to 1 this will be 1 so 1 and 1 will give you 1 and 1 on 1 will give you 1 right obviously but this 1 becoming 1 to be shown here as 1 will say take finite time to appear. Somehow or the other if this is let us 0 the output of this will be glitched as will going towards the 0 and therefore D will see this is the going to 0 and for the finite duration of the time F will show some 0 value and then it latch on to higher value. By this time your actually large amount of power dissipation has been lost here so it is always advisable and generally it is always recommended then try to make it as symmetric as possible.

(Refer Slide Time: 36:08)



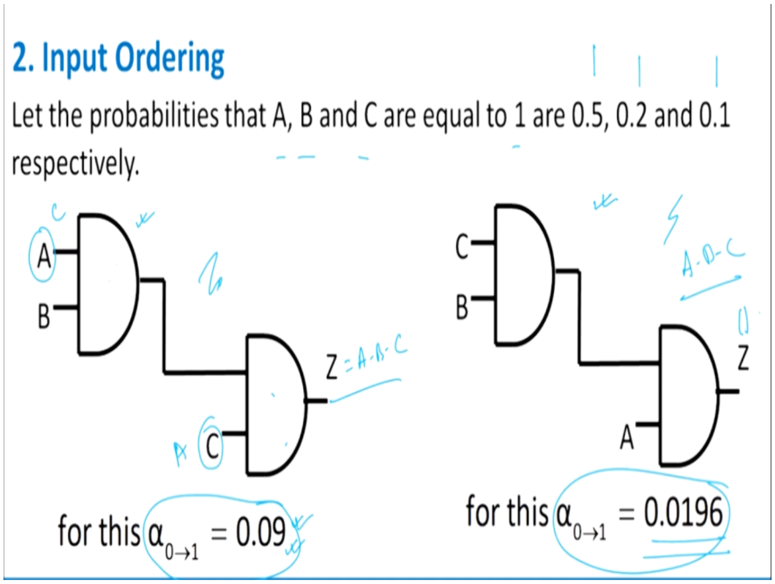
Design Techniques to reduce Switching Activity

1. Logic Restructuring

At that that option is basically in front of you which is this one. So when you makes something like this A1, A2 and let us suppose A3 then you ensure that the signal coming here and here and exactly is the same instant of time and therefore there is no problem in glitching and therefore your glitching activities reduced and your power dissipation are reduced drastically in this case right.

So this is one of the techniques design techniques for reducing power dissipation let me give you a concept of power input re-ordering right input ordering or reordering.

(Refer Slide Time: 36:47)



2. Input Ordering

Let the probabilities that A, B and C are equal to 1 are 0.5, 0.2 and 0.1 respectively.

for this $\alpha_{0\to1} = 0.09$

for this $\alpha_{0\to1} = 0.0196$

What we will learn here is how do I so I take suppose I take it this gate I also take this gate right but then the probability that A, B, C are equal to 1 if 0.5, 0.1 and 0.1 right so A as the higher probability of having 1 and C as the lowest probability of having 1 which is 0.1. So if you take this topology and if you take this topology tell us try to find out the value of 0 to 1 if you see this alpha 0 to 1 is 0.09 right 0.09 in this case.

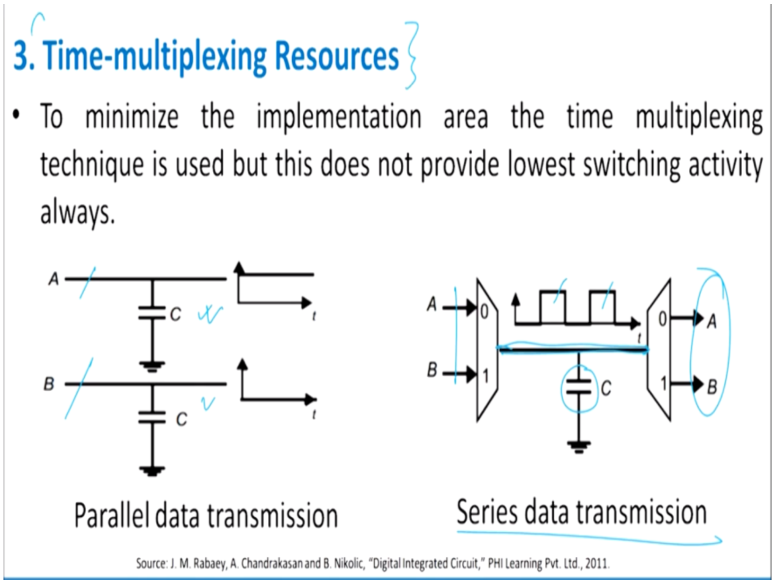
Whereas in this case exactly the same topology only thing which we have done is that A has been replaced by C and this C has been replaced by A right then in that case I get alpha 0 to 1 to be equals to 0.0196 right I get alpha to be equal to 0.0196 and it is relatively lower as compared to this one right. So simply by taking that input which bas got a higher probability of 1 available to it or higher switching probability I will automatically be able to and putting it closer to the output will reduce my activity effect right.

So this is one of the standard methodologies or one of the standard ways of saying that this methodologies is that if you want that your alpha should reduce let us have that input whose probability is higher towards one is more larger or larger probability you let it be introduced at a much later stage right.

So more you more late you introduced a highly high value of 1 which is in this case happens to be A lower will be your switching probability switching alpha 0 to 1 and therefore your lower will be power dissipation right and you see quiet interestingly in either of the two cases we have not actually sacrificed our methodology of we have not sacrificed the performance which means that your logic value are still remains the same if you Z will be goes to A.B.C here also Z will be equals to A.B.C fine.

So we have already one that this is something new in that in that sense right in that sense we come to what is known as time multiplexing sources what do I mean by that? I mean to say that let us suppose you have got a two types of data transmission.

(Refer Slide Time: 39:33)



3. Time-multiplexing Resources

- To minimize the implementation area the time multiplexing technique is used but this does not provide lowest switching activity always.

Parallel data transmission

Series data transmission

Source: J. M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuit," PHI Learning Pvt. Ltd., 2011.

I have got a parallel data transmission in these two buses A and B I have a serial data transmission where I do A and B and then I have a mugs here which allows it to flow from transmitter to the receiver and they will the these are the data point which is available to me and this interconnect will have a capacitance inductor and resistance so on and hence so forth and the output will be available to you at A and B.

So people thought was that let us suppose we do a time multiplexing what do I mean to say that I will introduce two signals A and B when A has got large number of zero's coming into picture I insert B is one into it somehow or other so your multiplex the value of B within A and you try to de-multiplex it at the output side right or multiplexing means putting one over the other so that they do not destroy any one of them and de-multiplexing basically means taking each other in time domain right.

And this is known as time multiplexing resources this technique well we would not be able to show too much at this level because it is much more of higher topic level at least for a bachelor course or a even a first year masters course this serial data transmission and parallel data transmission is not a very one of the good idea the first two were simply the best ideas available to us.

But this has been used quite a long time that if you are able to multiplex during the zero's or 1 of 1 bit A and your able to super impose on that the second bit cycle you will actually have a reduce alpha 0 to 1 value and therefore your power dissipation will be low in that case and so with this let me stop here now then when we start next time we will start with ratioed logic thank you very much.