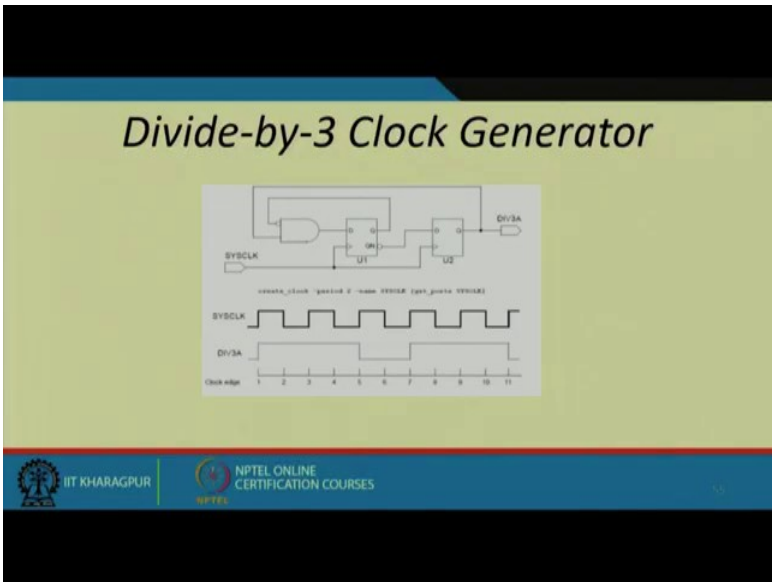
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture - 57
Timing Issues in Digital IC Design (Contd.)

Hello everyone, welcome to the course on Architectural Design of ICs. So, we are discussing mainly the Timing Issues which we found mainly in Digital IC Design. So, till now we have discussed there are several problems which we have to consider, to check the actual performance what will be the actual performance of the chip whenever it works in the real time. So, we have to consider all or we have to consider or constraint your clock with all the aspects of the original behaviour of the clock so that, we can get actual performance of the circuit while we are doing the simulation.

So, we have the we can generate the divide by clock to very easily by connecting the clock bar pin to the input port of a single registers which will be very easy circuit to divide or to generate one; that means, divided by 2 clock. So, by that configuration actually we can generate very easily the by 2 by 4 by 6 by 8 in the power of 2 divided by the bar of 2 clock generator very easily.

(Refer Slide Time: 01:20)



Now, in some case we may find that we require divide by 3 clock ok. So, on that particular aspect how we can define or how we can generate the clock 3 generator, not

which is not in the power of 2. So, at that time what we can do is that; suppose, we are having this system clock and divided by clock 3 means; at that time we are we need two different this registers where this the output of the first one and the output from the second flip flop, they are basically connected with a particular AND gate ok.

So, what is the meaning of this is that so, and their Q bar is basically connected to D. So, why it is divided by 3 whenever we are dividing by 3? So, at that time it is mainly this Q and Q bar ok. So, both if I say that the primary thing is that we have for divide by 3, we need some terminal count consideration. So, suppose this is the input system clock. So, whenever at this particular clock edge the Q is 0 and this Q bar is 1 ok. So, at that particular time this is 1 and this is 0 means this is also 1.

So, then it will this D input that will be just 1 ok. So, depending on that now it will try to go up to this. So, whenever we find that there is another edge ok. So, at this particular edge so, at that time what is this for at that time the output from this. So, earlier it was like the input from this is 1 so, at that time this will be 1 and then again this will be come down to 0 ok. So, 0 means here at this at this particular edge so, this will become 0. So, 0 will come over here and as this is 1 so, this will be also 0.

So, after; that means, at this particular edge, it will consider try to consider this up to this and in this particular edge it will come down to 0. So, why it will come down to 0 is that because, this particular gate has some delay. So, whenever this is coming after sometimes so, at that time at this particular age it will not find immediately that this particular that the output from this is 1.

So, it will after sometimes it can check and it can get or it can extend up to this and then you will get for this particular edge it will get the value as the here as 0 then again for 0 over here then again it will come and depending on this Q again it will come and then it will be just like 1. So, at this for 1, 2 and 3 this is 1 and then again this 0 and again 1, 2, 3 for 3 clock edge, it is 1. So, that is why this particular circuit will act as a divided by 3 clock generator ok.

So, the primary thing is that; initially as both are 0 so, it is starting with 1, then again as it is here it is going 1 means again it is 1 so, 1 and 1, it will start from 1. So, after that every clock edge it will just check the value and up to third clock; that means, this edge only it

will finds that ok. So, this corresponding D input is 1 so, it will be 0, then again 0 will come this clock divide there this at this the output of Q that will make 0 for that.

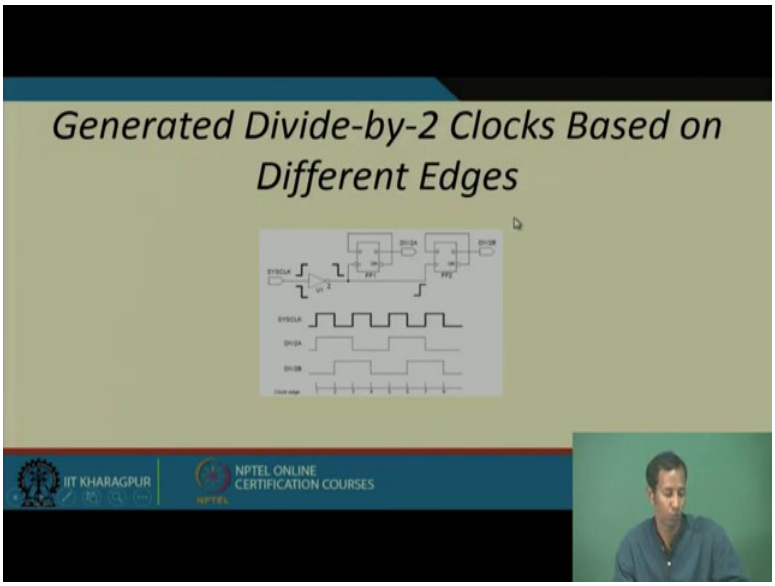
Again as this is coming as 0, so, 0 again it will affect this so, here if it is going 0 then again it will be just 1. So, whenever I will get 0 over here. So, at after 1 clock edge or after 1 clock cycles again it will make as 1. So that means, this is delayed by this 1 is continuing for particularly one cycles. So, in this manner we can or in this configuration we can make the divide by 3 clock generator. Now suppose we have to design one divide by 5 clock generator.

So, at that time what we have to do? So, there will be 3 registers at that time. So, divided by 5 means, I need 3 registers and the corresponding logic will be here as we consider this as let us say that 2 will be the terminal condition. So, whenever it counts this 2 over here so, at that time it will make the sequence to 0, then again 0 one two you count and then again if you will just make that as 0 or instead of that if you just make one FSM kind of things ok. So, there will be this 0, 1, 2.

So, if you develop from that excitation table you draw and then you if you come down to the logic of that it will just become the corresponding circuit ok. Now for divided by 5; what you have to do? Again you have to up to 4 you have to count; that means, 0, 1, 2, 3, 4. So, 4 of this edge will be counted whenever it finds that the four; that means, the 4 of this particular clock cycles has been counted. So, again immediately it will be come down to 0. Then again after for one clock period it will be just 0 then again it will be going back to the one.

So, something like that you have to generate or you have to you can create this divided by 5 divide by 7 divide by 9 divide by 11, any of this particular circuit related to this kind of configuration ok.

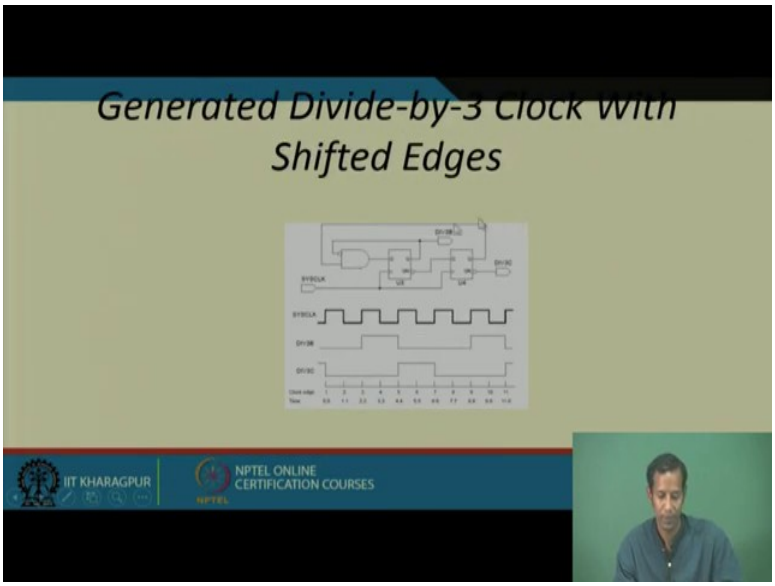
(Refer Slide Time: 08:55)



So, now, how you can generate divide by 2 clocks based on two different edges. So, suppose you are having this system clocks, now you just pass this system clocks through one inverter where this will be just; that means, negative unate clock. So, now you just this negative unate clock directly you just pass it to this one of this flip flop and another with inverted clock.

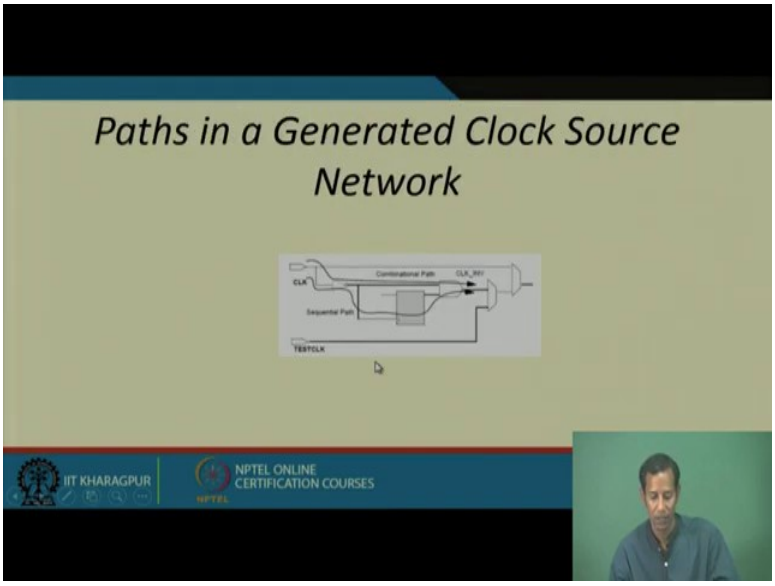
So that means; this will generate for this particular a flip flop, it will generate at the positive edge, it will generate the divider clock and here it will generate at the negative edge sense because here it is as it is already making as; that means, inverted. So, it will be just inverted; that means, at the negative edge at that time it will be creates the corresponding divide by 2 clock.

(Refer Slide Time: 10:00)



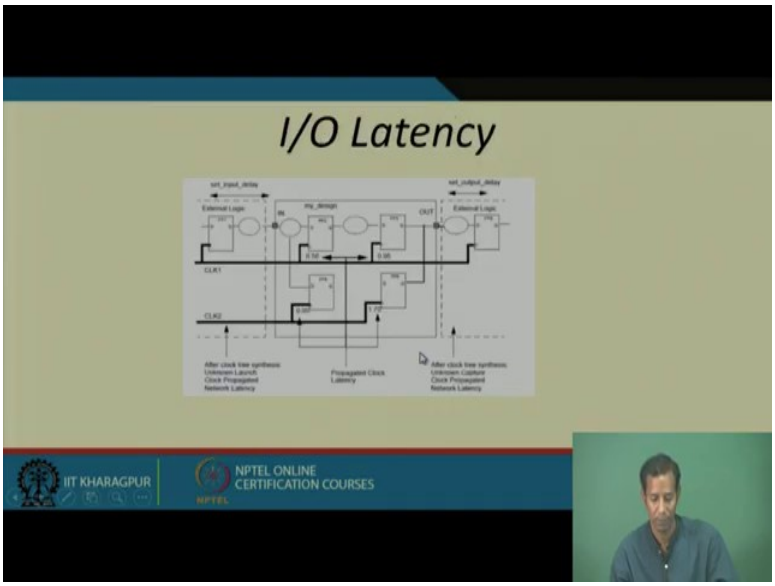
So, then again generate divide-by-3 clock with shifted edge. So, how we can do is that; now you are having this two actually two registers then again the same configuration and we where this particular actually this from this particular blocks output you will get the divide by 3 and from the corresponding Q bar output you will get divide by 3 and which is in this Q bar you will get shifted version of this particular which clock signals you will get from this shift register 1 or the flip flop 1 ok.

(Refer Slide Time: 10:48)



So, then the paths which are there in the generated clock source network is that you will be having one combinational path, you will be having one sequential path and in this clock signals will be just connected to this sequential elements passing through some inverters or some buffers.

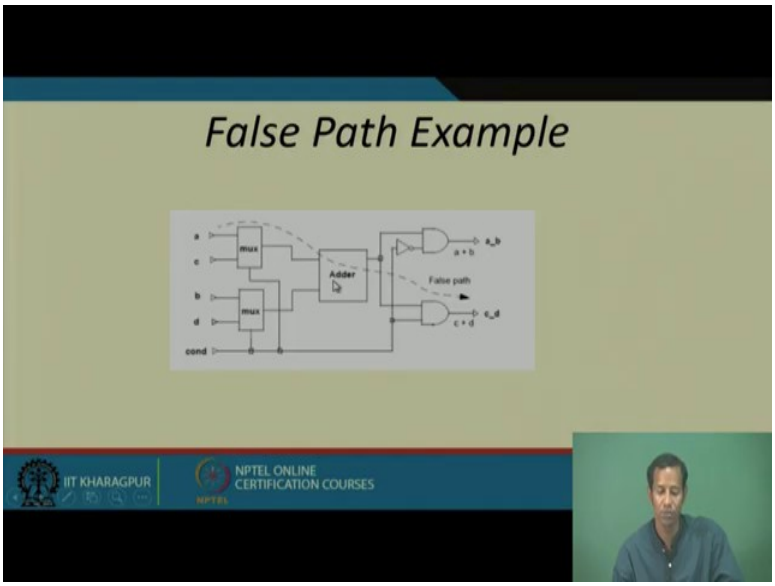
(Refer Slide Time: 11:11)



So, this I O latency we have already said we already have seen that. So, suppose this is my block, this is my design and this is the external block which is connected to my design so, this is the input delay this is the amount of input delay to my design and if my block is again connected with some other blocks.

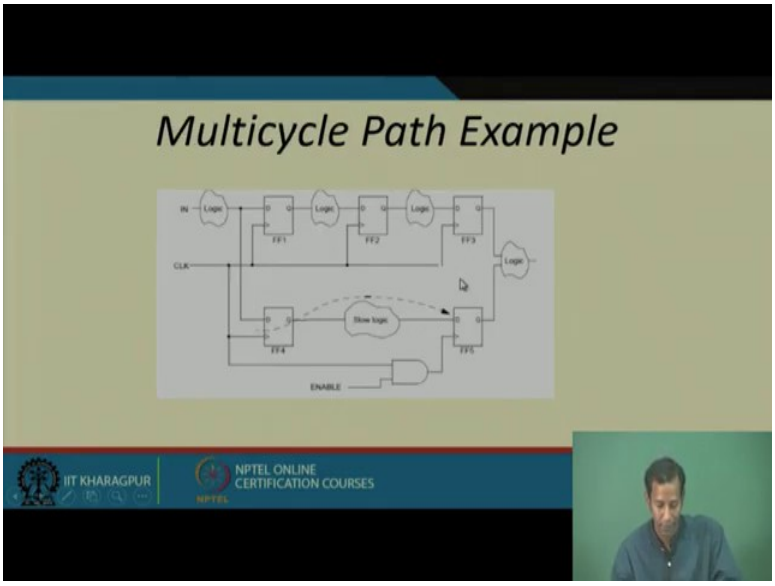
So, then what will be the amount of delay requirement from reaching from this particular output to the input of the other blocks. So, that is the amount of output delay which you have to consider while we are designing our circuit.

(Refer Slide Time: 11:49)



So, then this, what is the false path example is that? Suppose as we are having a b c d and we are having this condition. So, depending on this condition as there are; that means, all the paths are basically connected from a to b or c to this d d b to this d to this, but this particular path will be never activated. Depending on the logic this particular path will be never activated which will follow that a to this. So, that is why this path is named as or this path is called as false path.

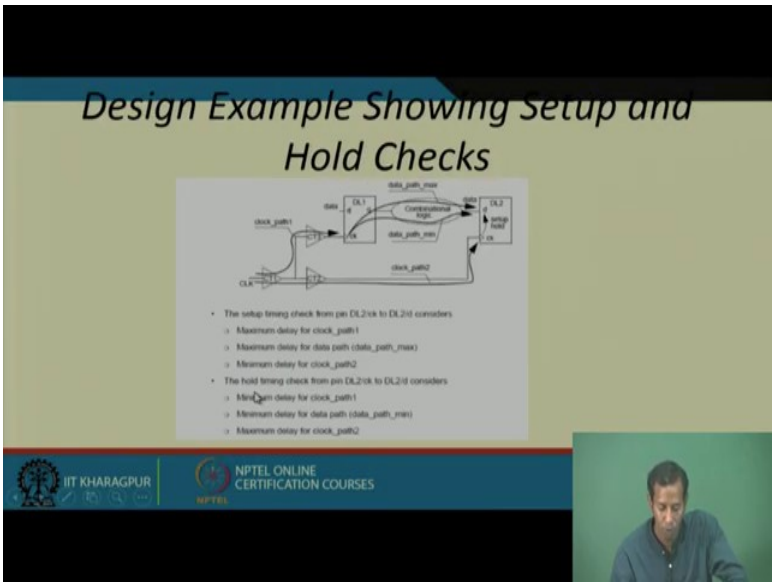
(Refer Slide Time: 12:31)



So, then what is the example of multi cycle path? Multi cycle path is that suppose, we are having some logic something like this; that means, we are having logic then flip flop logic then flip flop logic then flip flop. Then again here in this we are having some slow logic. Slow logic means; suppose, here in this all these are basically let us say let us consider these are connected with full adders. Here inside of these two flip flops we are having one this memory cell sorry one this multiplier cell or multiplier logic.

So, the multiplier logic requirement delay requirement will be much more than this full adder cell. So, which will basically though the requirement timing requirement for this logic is much smaller so, depending on this corresponding multiplier logic the overall design cycle time will be much reduced. So, that is why if we if you assign that this particular logic require multi cycle to complete the whole operation. So, at that time this path has to be considered as multi cycle path ok.

(Refer Slide Time: 13:51)



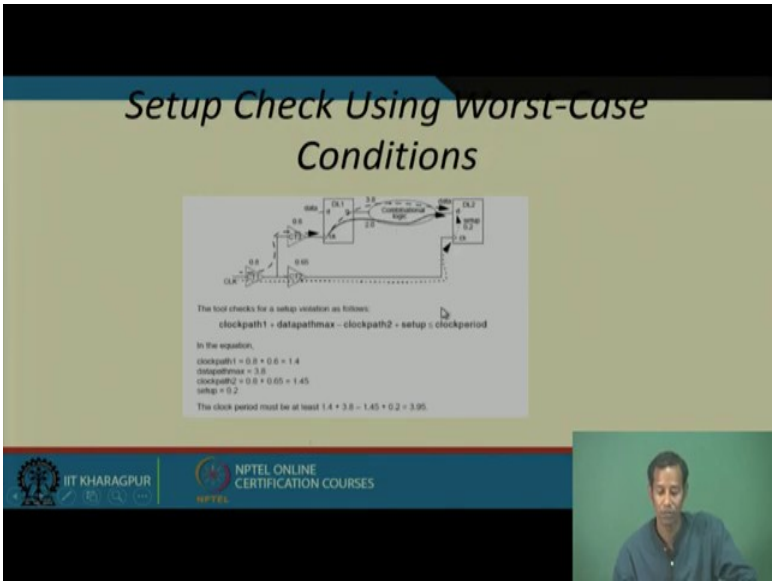
So, how we can in any of this example, how we can actually check, in timing analysis tool or actually using some mathematical modelling how we can check what is the; that means, set up time violation or set up time requirement or what is that that means hold time requirement.

Suppose we are having some design configuration something like this. So, the set up timing check from this is the this flip flop 2 flip flop 2 s clock pin to the flip flop 2s d pin we will consider the maximum delay for clock path 1, this is a clock path 1, then the

maximum delay for data path; this is a maximum delay for the data path as well as the minimum delay for the clock path 2 as well as the minimum delay for the clock path 2.

And the hold time checks for that; that means, this dl 2 to this the input of this we will consider the minimum delay for this clock path 1, then the minimum delay for data path 1 and the maximum delay for clock path 2 ok.

(Refer Slide Time: 15:09)



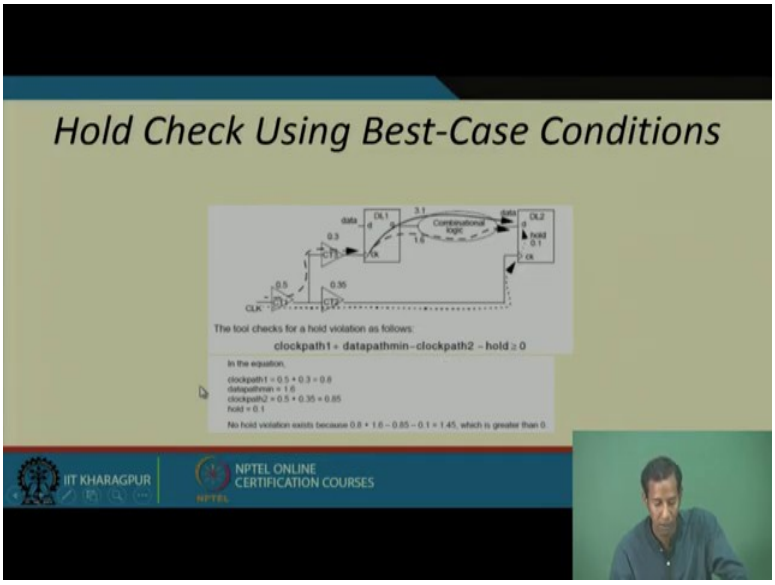
So, all this it will consider so, depending on this if I first let us check what the setup value requirement for this particular case is. So, the tools we will check for the setup violation or manually also I can check if I know all these values delay information of all these values. So, at that time this the setup requirement is said that it will check for the maximum delay for clock path 1, maximum delay for data path and the minimum delay for clock path 2.

So, the clock path 1, then the data path max minus clock path 2 plus set up less than should be the clock period. So, in this section; that means, in this particular circuit we are having this clock path 1 is this 0.8 plus 0.6 which is the summation of C t 1 plus C t 2 sorry, C t 3.

Then the data path max which is this path is 3.8, then the clock path 2 is 0.8 plus 0.65 so, 1.45 and the setup requirement for the flip flop is 0.2. So, the clock period must be at least this clock path 1 plus data path 1 minus this clock path 2 plus this setup which is

this 3.95. So, this will be the setup time requirement of this or I can say that the minimum clock period should be set for this will be 3.95 to meet the all constraint in this particular circuit.

(Refer Slide Time: 16:56)



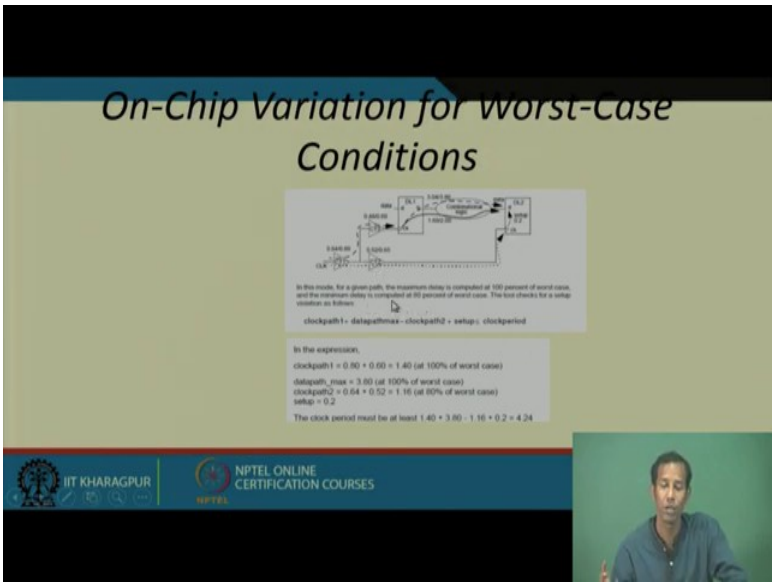
Now, for the hold check what will be the consideration for the same circuit. So, not same, actually here that here the; that means, the parameters are changed so, the configuration is same. Now for hold actually for hold what we have to consider the minimum delay for clock path 1, the minimum delay for the data path and the maximum delay for the clock path 2.

So, the equation is that clock path 1 plus data path min minus clock path 2 minus hold. So, in this equation the clock path 1 is this; 0.5 plus 0.3 which is 0.8 then the data path is basically the minimum of the data path is 1.6, the maximum is 3.1, the minimum is 1.6 ok, 1.6 then the clock path 2, this clock path 2 is 0.5 plus 0.35 which is 0.85 and hold requirement is 0.1.

So, then there will be no hold when the corresponding equation 0.8 plus 1.6 minus 0.85 minus 1, this is 1.45 which is greater than 0. So, that is why this design will not consider any of the hold violations. So, now, if you consider; that means, if you get this value as less than 0. So, at that time you will find the hold violation.

So, when you will get the hold violation? Means what, the data reaches too early; that means, if this particular value is let say 0.1 so, at that time you will get this value as point minus 0.05. So, minus 0.05 means that much of hold violation you will get in this particular circuit.

(Refer Slide Time: 18:54)



So, this on chip variation for worst case condition, then again this setup actually you can consider this clock path 1 here actually on chip variation means you have to consider the minimum and the maximum both depending on the this process voltage temperature variation condition.

So, here you are considering that for worst case the delay is 0.8 plus 0.6. So, 1.4 that is 100 percent of the worst case and in worst case the data path delay is 3.8. So, here it is 3.8 for the clock path it is 0.8 plus 0.65 which is 1.16 sorry, not actually for this we are considering the minimum sorry minimum delay we have to consider. So, here we have to consider the minimum delay for the clock path 2.

So, minimum delay for the clock path 2 is this 0.52. So, 0.52 plus 0.64, that is 1.16 and setup is 0.2. So, the minimum clock period requirement at least will be 1.4 plus 3.8 minus 1.16 plus so, 4.24 which will be the same case. So, in this particular case we are considering only single case, but in this case we are considering as well as best case as well as worst case; that means, if particularly this information this delay information is

already given to me then only I can say or I can use them to calculate what will be the time requirement for setup or hold in any of this circuit ok.

And here actually, manually only you can do as you are having only two registers and you are having only one or two paths so, that is why it is much easier. But whenever you will work on this complex circuit design so, at that time you might find that there will be millions of paths which consists something like this configuration. So, at that time each of these paths you have to calculate. So, each of this calculation manually it is not possible to do.

So, that is why at that time what you have to do? You have to based on this equation the tool that the EDA tool or the electronic design automation tool they calculates and it shows that you are getting each of the paths they calculate and it shows that you are getting you are having this much of paths.

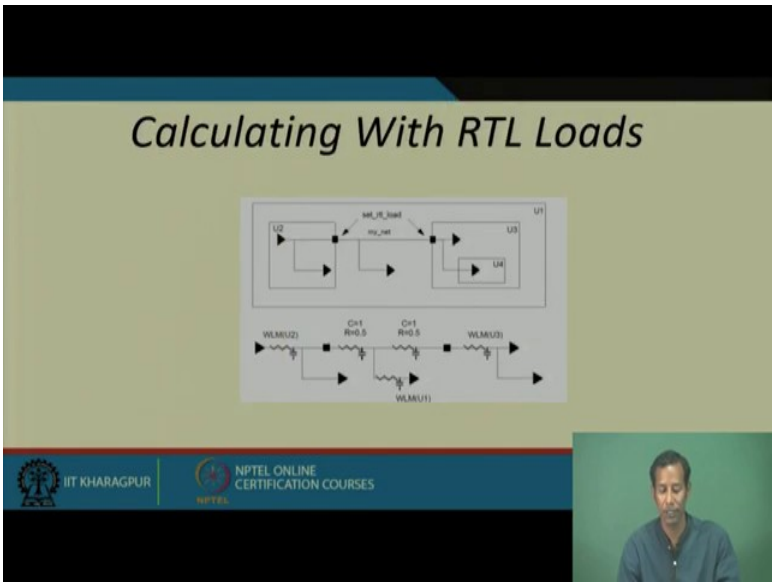
So, among them if there is any violation so, which path are getting with how many violation that will be shown to you if you are not getting any violation. So, at that time with how much slack you are getting or what is the results with the slack that will be indicated in the timing summary report of the whatever tool you are using as the static timing analysis of your circuit.

So, here manually you can actually this is just as an example that how you can calculate each of these paths resist; that means, the timing analysis value. Now if you are suppose from the circuits aspects suppose you ran the tool for timing analysis and then you see that for some paths you are getting the result; that means, you are getting some of the violations.

So, at that time you can identify that path and then again you calculate manually that this is the information about the delay now what changes should I make. So, that I can overcome the problems of this violation and my chips work perfectly fine ok.

So, for that reason you have to know how we you have to or what particular aspects you have to calculate for hold check as well as the timing check while you are doing the timing analysis.

(Refer Slide Time: 23:02)

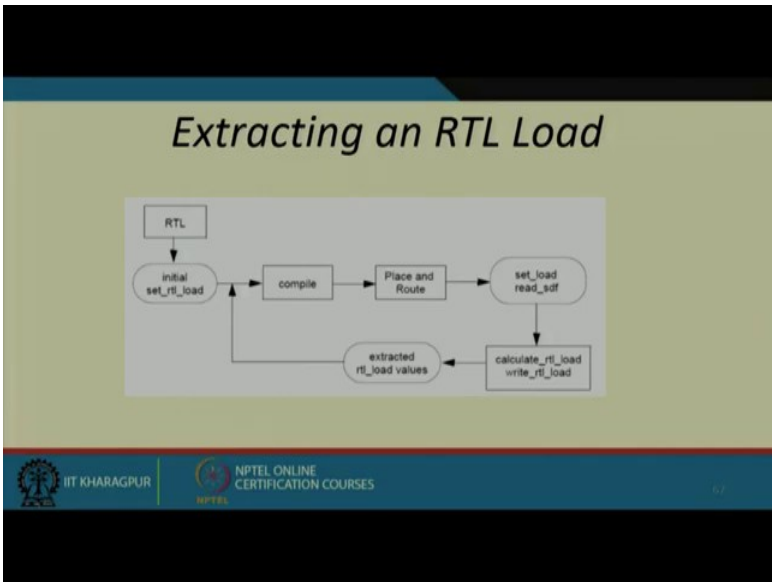


So, now, how to calculate this or how this tools basically calculates the corresponding this net delays is that, suppose this is one U 1, U 2 is one of the block U 3 is another block now they are basically connected with this.

So, these are the nets, now all this is connected like this wire load model. This is a wire load model of U 2, this is the wire load model of this particular nets. So, which corresponds to some of this capacitance value as well as this resistance value then against this resistance value, now again it is connected with this another wire load model which is going to this block U 1 and then it is going to the wire load model of block U 3.

Now based on this now the spy simulation will run in the background and it calculates that for this configuration or for this capacitance and the resistance value of this what will be the corresponding delay for this particular path. It calculates and depending on the corresponding loads this value changes the capacitive load changes. So, depending on the calculation the value will be reflected on the timing analysis report.

(Refer Slide Time: 24:29)

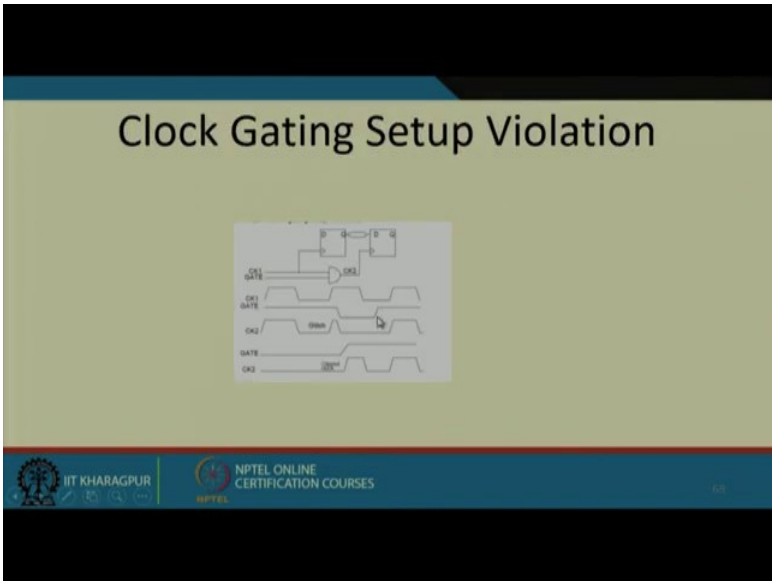


Then this is the actually how we in physical design this is the; that means, steps which is followed for the timing analysis. You are having the RTL so, then initially this we have to set the rtl load, then you compile, you do the place and route and then you read the corresponding standard delay format sdf file and then calculate the rtl load write the rtl load and then again extract the rtl load values extracted means all these this c and r the depending on the wire load model you have chosen.

So, all the c and r values you have to extract and then again it will be back annotated and it will be calculated depending on the whatever rtl code you have written. So, it checks the timing you have suppose we have set the timing my clock period requirement is let us saying 5 nanosecond. So, it checks it performs all these steps and then again all the delay information is calculated and then again it corresponds to your that your required time. So, you are if you have said that is 5 nanoseconds. So, that is your required time.

Now, the actual time will be calculated based on this particular loop. So, actual time if it is more than 5 so, at that time you will get negative slack if this particular this path corresponds to the lesser value than the required time. So, you will get positive slack ok.

(Refer Slide Time: 26:09)

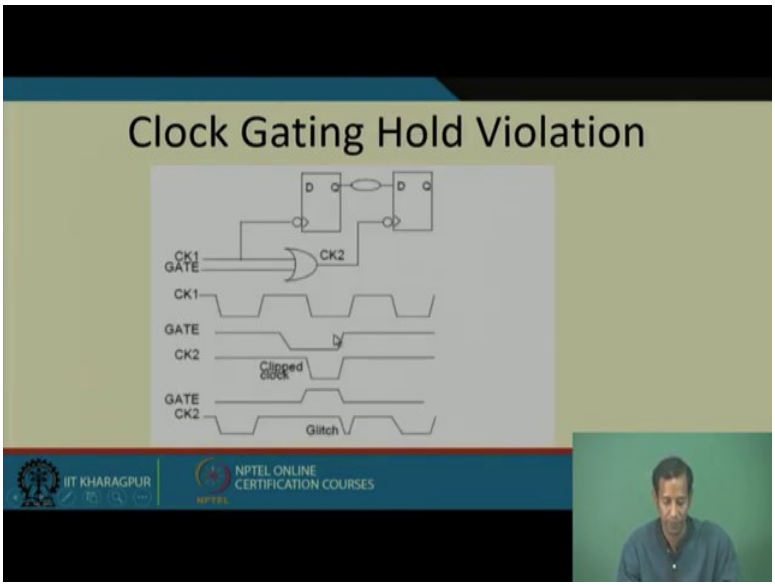


So, this is so, actually I was saying that in clock gating you will get sometimes you will get the glitch. So, if I say that this is the clock 1 and this is the second flip flop is basically getting the clock gated one which is the clock 2 signals so, clock 1, now the gate signal if it is something like this. So, at that time then the clock 2 will be here as this is 1 this is 1 so, it will be 1, but because of this delay in the clock 1 and the gate you will get some of the glitch over here.

So, when you will not get the glitch over here is that if the gate signal is something like this at that time there is no such glitch at the clock 2 and here if you are getting any glitch means what the corresponding circuit performance will be totally wrong ok.

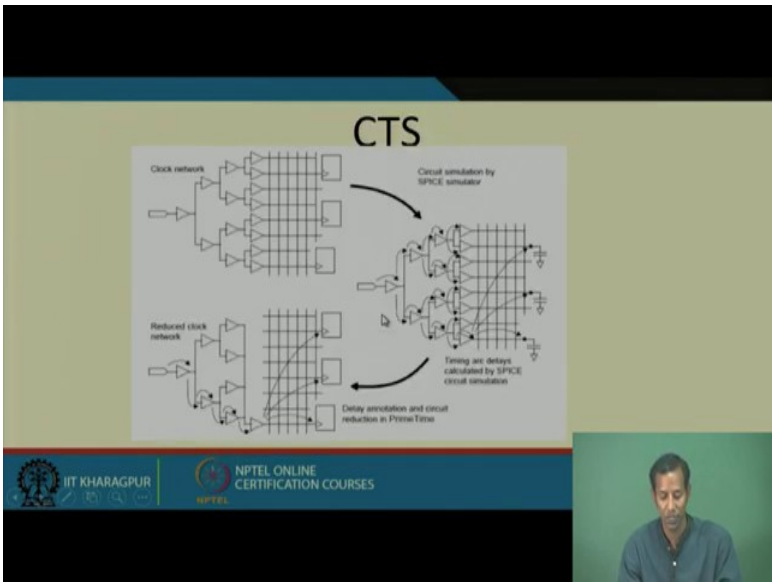
So; that means, if you are having glitch, glitch is the unwanted signal which we got in our desired signal. So, whenever it finds one so, at that time the corresponding data unwanted data will be sampled in the second flip flop ok. So, that is why whenever we are doing this gated clock circuit design at that time we must have to be very much careful about so, that this glitch may not occur at the circuit ok.

(Refer Slide Time: 27:36)



So, then again actually we are we can get some of the; that means, this is some of the configuration where we can get because of the hold time we can get the glitch at the clock 2 and that can create a problem in your circuit.

(Refer Slide Time: 27:55)



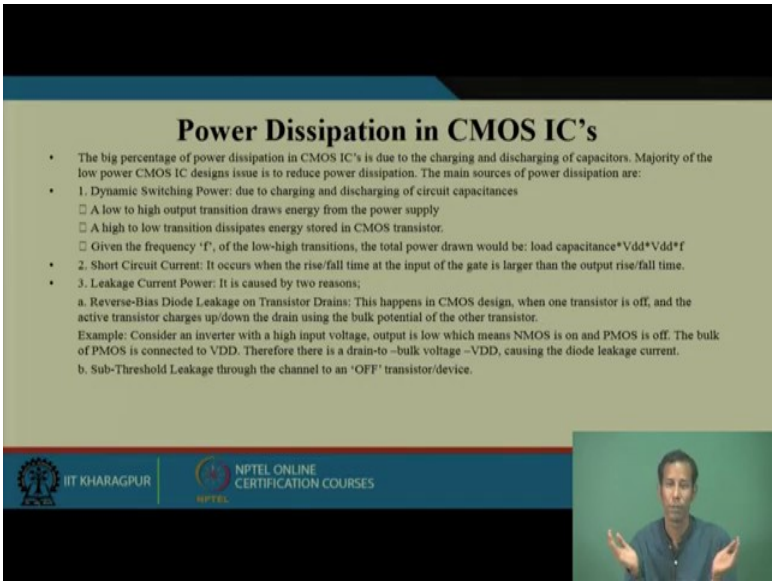
So, this is this clock tree synthesis so; that means, this is the clock source pin then it is connected as H tree ok. So, you see from one buffer then again it is connected with two buffers then again two then again it is distributed two and again by this will create the

network and it will connect it each and every sequential element which are present in your particular circuit ok.

So, then this particular circuit will be extracted through the spice simulator and it will try to optimize because here we are using; that means, as this is 1 then from 1 it is connected with two then 1 then again with this connected with two.

So, it creates this particular type of mesh for the clock network. Now as these all of these requires the buffers so, we have to minimize the number of buffers how we can minimize? So, through the spice simulation it tries to minimize the number of buffers which is used for these particular cells and it come downs to a lower number of buffers which will be used for the reduced clock network ok.

(Refer Slide Time: 29:13)



So, in actually CMOS IC design we are having different kind of source of power dissipation. So, you will be having this dynamic switching power consumption, you will be having this short circuit current power consumption, you will be having this leakage current power consumption so, all these are there.

So, if you are working on this low power digital VLSI design. So, at that time you have to consider your design aspects in such a way. So, that the power consumption in all these aspects should be in the lower domain ok.

So, well this is it for actually this timing issues or the timing information ok. Next we will see that optimization what we can do the optimization in the circuit. So, that we can get the circuit performance in the higher side in terms of speed power and area.

So, thank you for today.