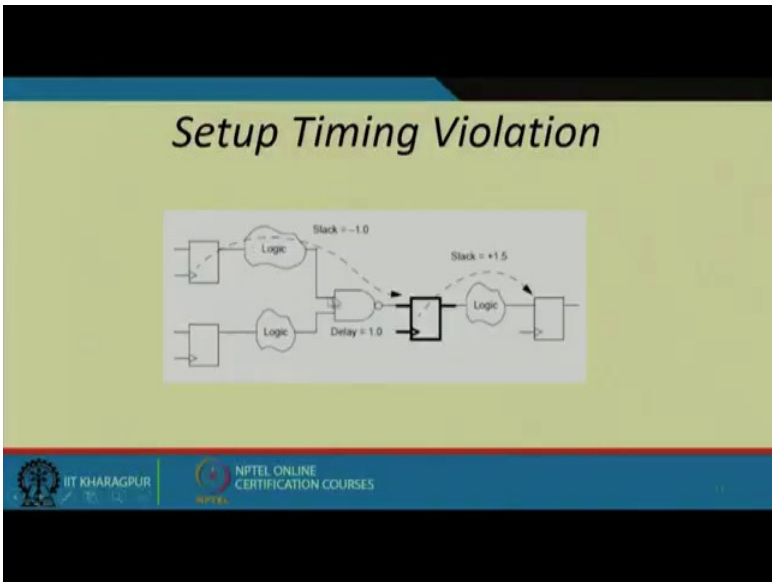
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture - 56
Timing Issues in Digital IC Design (Contd.)

Hello everyone, welcome to the course on Architectural Design of ICs. We are basically discussing the Timing Issues in Digital Integrated Circuit Design. So, we have seen that we have to consider all these clock latency, clock jitters, clock uncertainty all these things we have to consider while we are doing the timing analysis. And we have also seen that as a designer whenever we are partitioning the design at that time how it affects the timing analysis in further ok.

So; that means, if we design their circuit in a; that means, good manner at that time. We can analyse this that performance of the circuit timing in a very good way or if we have done poorly designed circuit. So, at that time you might get some of the problems while you are working with the complex circuit. So, today we will see this one of these set up time violation ok. So, what is set up time violation in this particular circuit, if this is one of the examples where we can find this set up time violation.

(Refer Slide Time: 01:32)



So, in setup time violation if you see that, there are some registers and in between this registers there are some logics which are placed. So, we are having two different paths

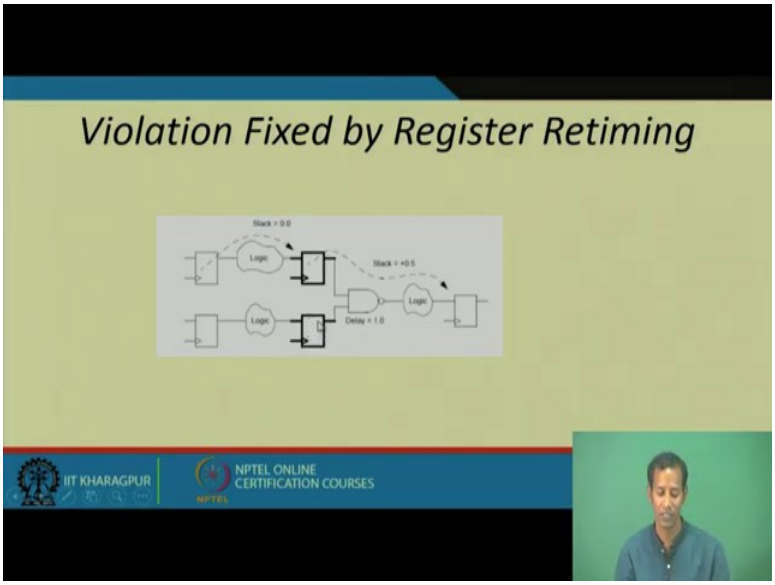
which are NANDed over here and then it is passes through one registers then again there are some logic and then again it is passing through one-th register.

So, in timing analysis what we have found that in these particular paths it is getting the slack of negative one. So, the slack of negative ones means, the actual arrival time and the required arrival time they have the difference of 1 minus 1, but in this particular case you are having the surplus; that means, the slack is in the positive 1.5 amount of time.

So, what I said is that if your; that means, required time minus the actual arrival time they are very much closer together. So, at that time your slack value will be if they both are same. So, at that time slack value will be 0. So, in a perfectly optimise circuit the slack should be 0 or near to 0. If you are getting any negative slack means the design will not work perfectly fine because, there the actual arrival time is much higher than the required time. So, required time at that time you have to increase the required time to; that means, meet the actual arrival time which is more ok.

So, which happen in this particular case. So, is there any technique without putting or without doing anything; that means, you not doing anything, but with a; that means, not doing anything means with without not putting extra hardware or extra logic to it, can I just by reshuffling or by some changes can I make this circuit works perfectly fine though it is having the set up time violation.

(Refer Slide Time: 03:42)



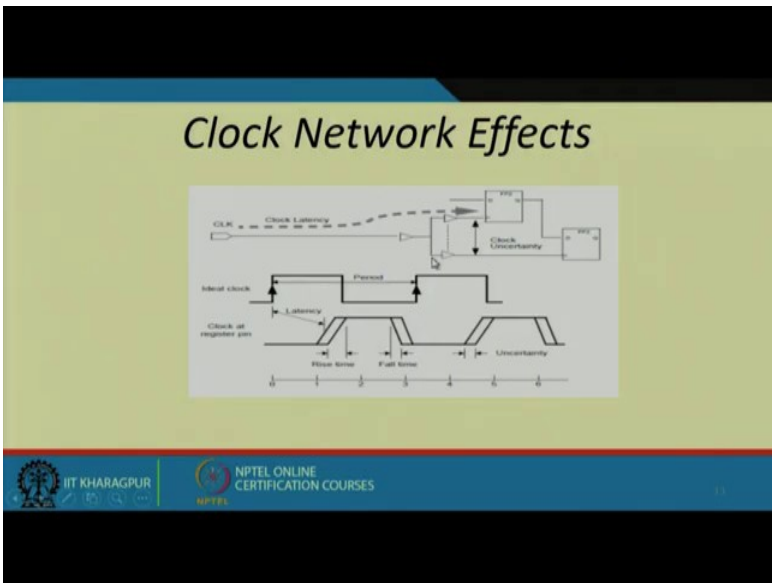
So, if I just make that this particular registers I just place here and there and then I just; that means, the NAND gate if I place just after this registers not before the registers.

So, at that time what will happen now the delay of this particular NAND gate is 1. So, whenever I place this to this. So, at that time the slack requirement will be minus 1 to 0.5 and here it will be getting slack of 0. So, I just place the registers over here in this path and then NAND, then the logic and then there is register.

So, as this particular NAND gate cells corresponds to the delay of 1 time period. So, now, the modified slack in this path that will be point plus 0.5 and slack in this path that will be 0. So, in this path the slack will be plus 1 ok, but what we have put we have just put the delay and we have taken out this registers just; that means, before the NAND gate.

So, we have to copy the registers in this two particular path. So, we have to put one extra registers in this particular path to achieve or to fix the violation which is creating the problem in this case. So, this particular technique is known as register retiming ok.

(Refer Slide Time: 05:23)



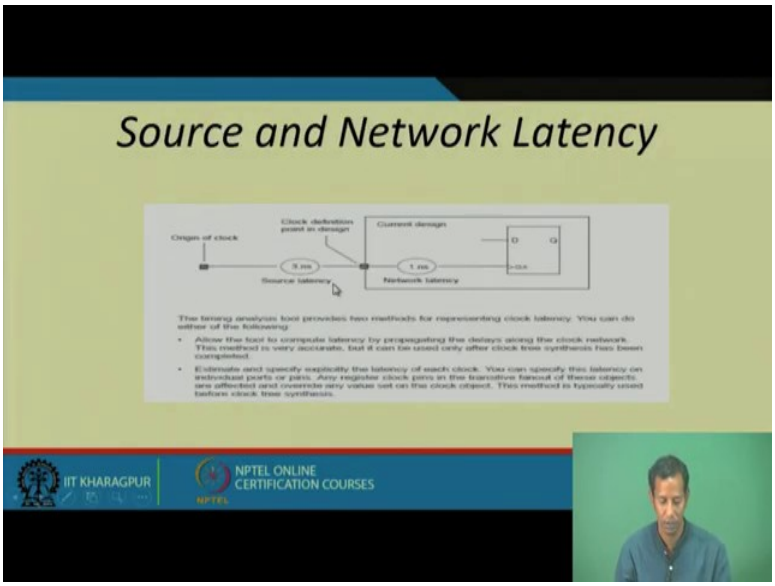
Then again what is actually what I said is that whenever we are having this clock source and they are connected to different sequential elements so at that time; we have to consider the clock tree network or in digital IC you have to create or you have to consider the clock tree network.

So, they clock tree network that may be that there are types of actually clock tree network. So, from this clock pin to the corresponding elements of this; that means, the flip flops clock means if it is connected this is a clock source pin and this is the pin where this registers clock is connected. So, if the corresponding time requirement from clock pin; clock source pin to the clock pin where the where it is actually connected to the sequential element this is known as the clock latency.

And as there it is connected here through this buffer and here through this particular path, they may be as their length is different so the corresponding clock edge will not be same. So, that is why this particular difference between the clock edge that is known as clock uncertainty. So, here you see this is the clock edge difference. So, that is the clock uncertainty. So, depending on this clock network effects we can get this rise time, fall time as well as this clock latency and the clock uncertainty. So, whenever we will consider or when we; that means, we generate the corresponding layout of your circuit.

So, at that time for the clock pin we have to or for the clock source we have to consider we have to constrain the tool by mentioning every of this aspects of the clock. So, that the timing information becomes closer to the original one which will be the real time performance of your circuit ok.

(Refer Slide Time: 07:39)



So, then the source and the network latency. So, this is the origin of the clock origin of the clock means this is the, if I say that this is the clock pad ok. So, then from clock pad

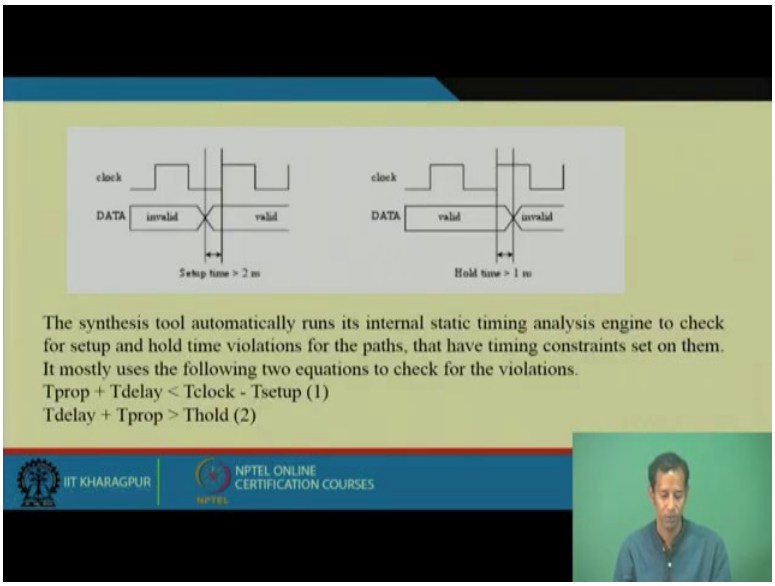
to the corresponding clock pin this is the core ok. So, this is the clock pad sorry clock pin.

So, clock pad to the clock pin you might be having some source latency then from source latency to the element where this clock is connected they will be also having some of the latency which is known as this network latency. So, the latency is basically divided the clock latency is basically divided into two category; one of this is source latency and the network latency. So, you have to while you are defining or you are providing the information about the clock latency at that time you have to consider this source latency as well as the network latency both as the summation of this as the clock latency to your circuit ok.

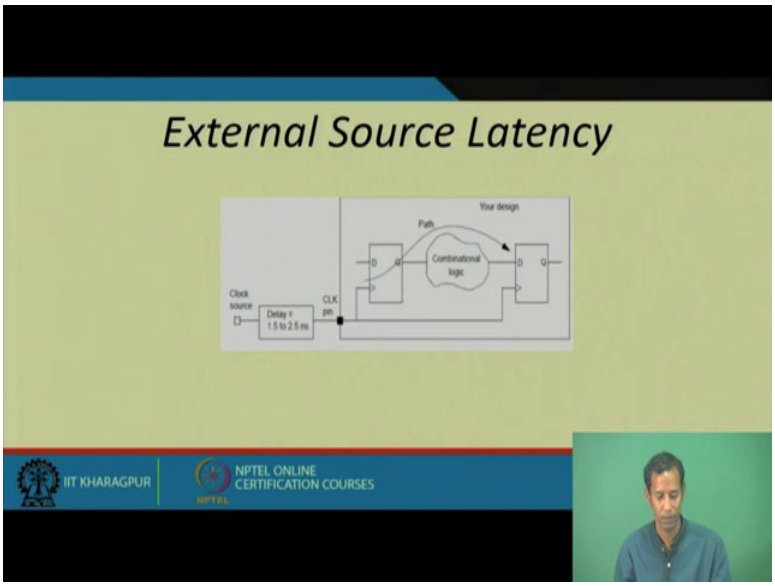
So, the timing analysis tool provides two methods for representing the clock latency you can do either of the following allow the tool to compute latency by propagating the delays along the clock network this method is very accurate, but it can be used only after the clock tree synthesis has been completed. And estimate and specify explicitly the latency of each clock, you can specify this latency on individual ports or pins and any other registers clock pins in the transitive fanout of these objects are affected and override any value set on the clock object this method is typically used before clock tree synthesis; that means, if I mention that manually.

So, that is done before clock tree synthesis and if you; that means, whenever you will do in that clock tree synthesis in physical design; other aspects of your VLSI; that means, digitalized integrated circuit design. So, at that time the tool automatically will find out what clock latency it is achieving and depending on that it will calculate the corresponding timing analysis.

(Refer Slide Time: 10:11)



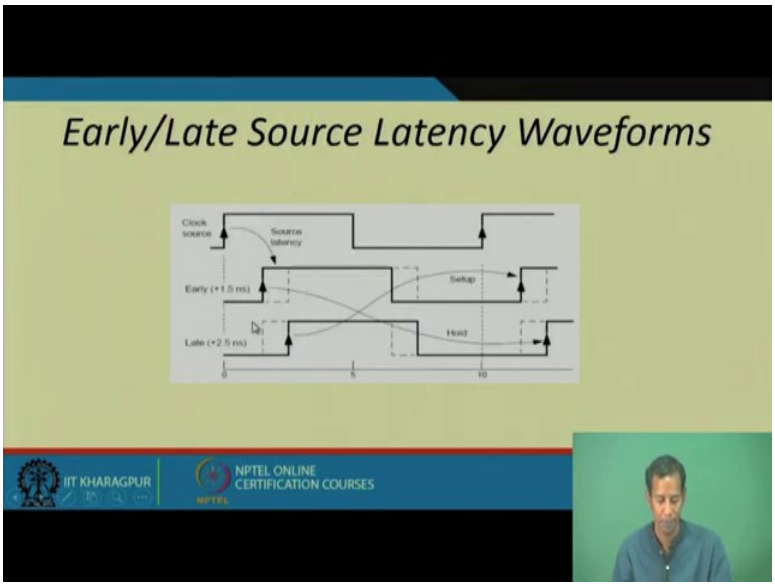
(Refer Slide Time: 10:14)



So, then actually we have already seen. So, when external source latency ok. So, external source latency means again we are having that from the clock source that source may be inside of the chip or outside of the chip, there may be some of the delay to achieve at the clock pin of your core ok.

So, this is your design or your core which is basically considering this clock pin.

(Refer Slide Time: 10:55)

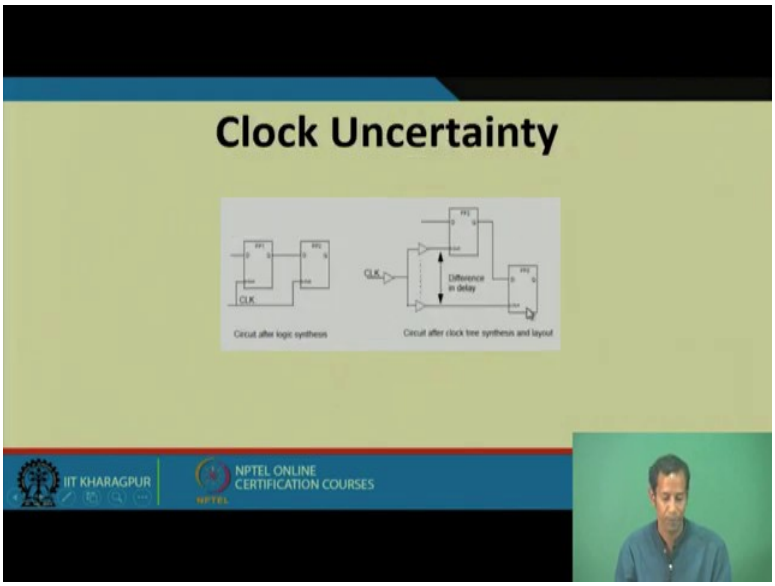


And you might be having the delay you can also define or you have to consider that latency as the source latency to the clock pin ok. So, now, then if we consider the corresponding waveform of this considering this early or late source latency; that means, consideration. So, at that time if this is my clock source ok.

So, early means what early means actually you will be getting the clock; that means, after let us say; that means this depending on this delay of this clock latency the corresponding edge will occur here which is 1.5 and for 2.5 nanoseconds of.

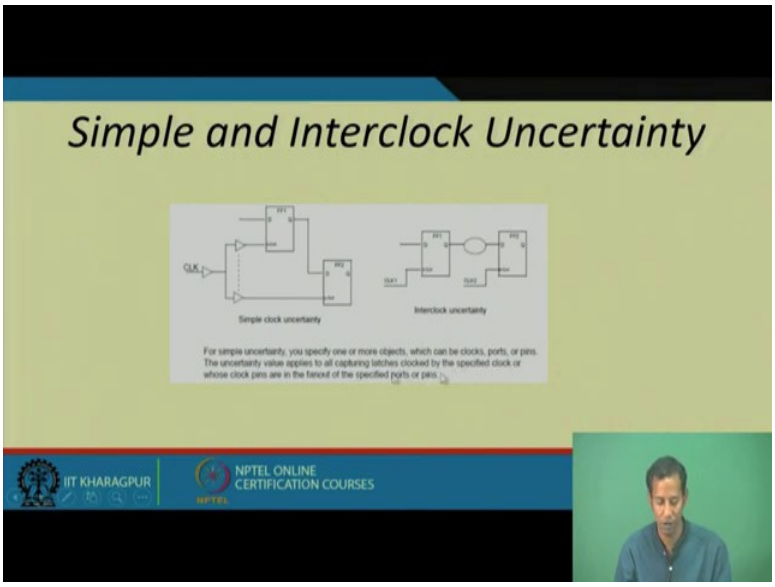
That means this latency you will be getting the corresponding edge at after 2.5 nanoseconds of time ok. So, now, at this particular edge you can check the hold at this particular edge you can check the setup ok.

(Refer Slide Time: 12:03)



So, we have already seen what this is. So, this is the circuit after logic synthesis and whenever this same circuit will act or it will behave as this one whenever you do this clock synthesis in the layout part.

(Refer Slide Time: 12:22)



Now, there are two type of actually uncertainty; one is this simple uncertainty and another one is inter clock uncertainty.

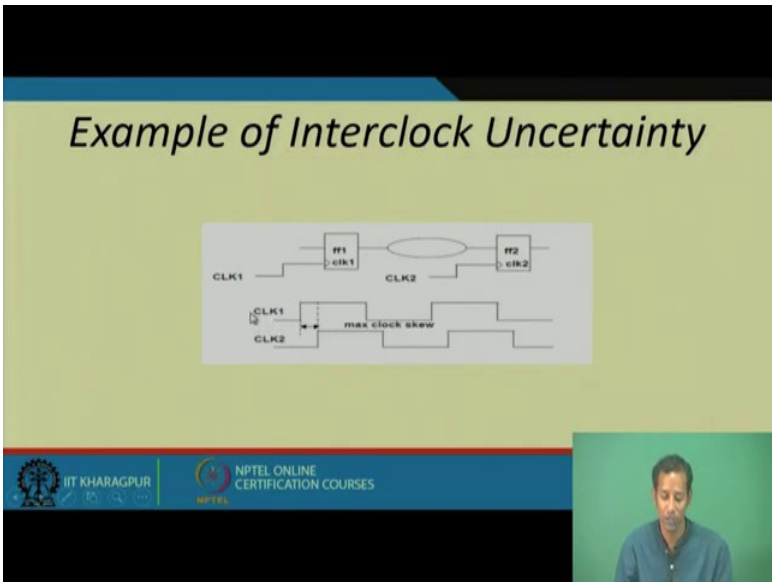
So, the simple clock uncertainty is what whenever we are having this clock source and again it is connected with this different flip flops with the buffers. So, that this one is a

simple clock uncertainty and inter clock uncertainty is that suppose we are having two different flip flops are connected with two different clock source.

So, at that time and then if they are having some combinational part in between of these two registers; that means, the two clocks if they are asynchronous to each other. So, at that time we may get this inter clock uncertainty ok. So, we have to consider the at that time for better timing analysis we have to specify that these two clocks are asynchronous to each other and they are having this uncertainty value of each of the clock is like this and like that depending on this the source of clock 1 as well as the clock 2.

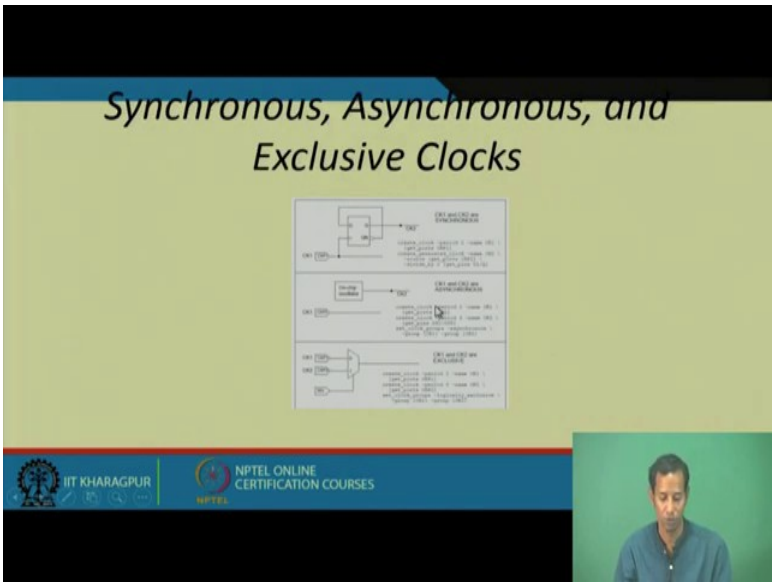
So, for simple uncertainty you specify one or more objects which can be clocks, ports or pins. The uncertainty value applies to all capturing latches clock by the specified clock or whose clock pins are the fanout of the specified ports or pins ok.

(Refer Slide Time: 13:57)



So, now whenever we are having this inter clock uncertainty. So, at that time you may you may seen that ok. So, the clock 1 is basically arriving or it is something like this the clock 2 is; that means, the edge of the clock 2 they are having some difference ok. So, that is the maximum clock skew and you have to find out or you have to mention that depending on this the changes in the edge of this clock 1 and clock 2 you have to specify what is the amount of this uncertainty or of this two asynchronous clock group which is clock 1 and clock 2.

(Refer Slide Time: 14:37)

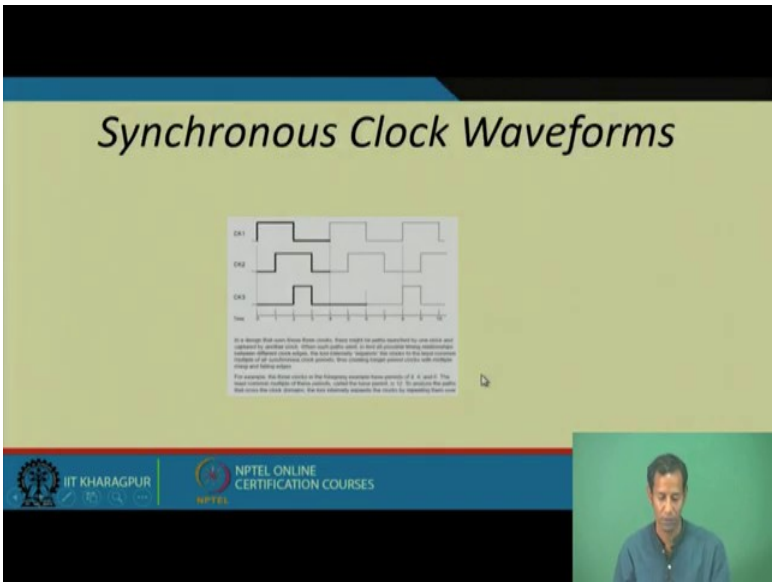


Then we are having a different type of clocks which will be applied to our circuits. So, the clocks which will be applied or which will be used in the; that means, design they may be synchronous or they may be asynchronous or they may be exclusive clock.

So, the synchronous means actually all the sequential elements will be connected with the single clock and if you are having different like clock 1 then clock 2 different clock more than one clock in your system. So, at that time they will be asynchronous to each other and if you are having this exclusive clock means you are having one that clock 1 and clock 2. And depending on this select you can select it will pass either of clock 1 or it will choose either of clock two.

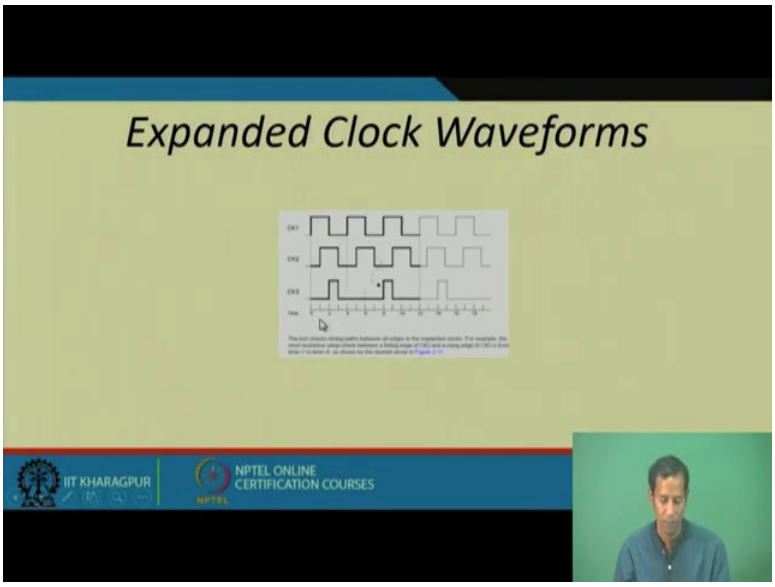
So, one of the example I is like that if we are having this frequency synthesizer. So, at that time we need different clock should be a go to the output and it will go to the different blocks ok. So, at that time. So, different clock frequency requirement itself makes the design that the clock source as the exclusive or ok.

(Refer Slide Time: 15:54)



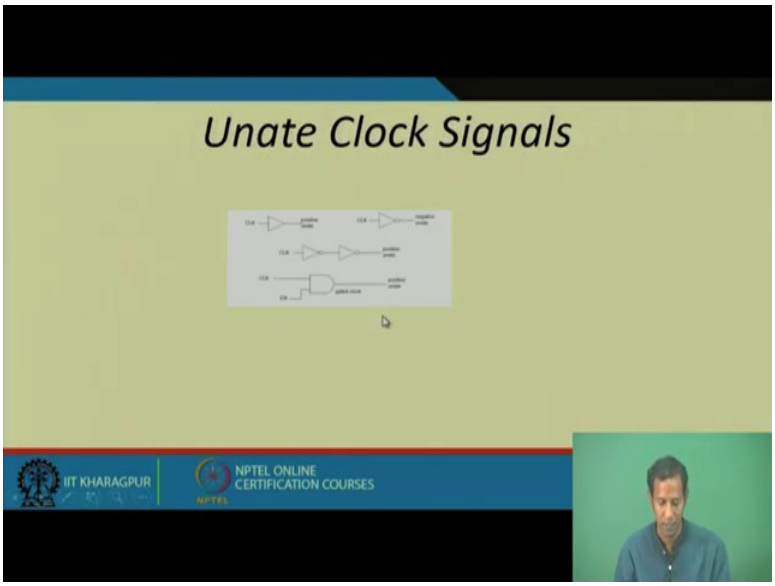
Then this is the synchronous clock waveforms suppose this is the clock 1 then see this is the clock 2 and now, if I just clock 3 is the exclusive or of this if we do then at that time not exclusive or clock 3 is a I think there is a different clock.

(Refer Slide Time: 16:15)



So, then this is another example of expanded clocks.

(Refer Slide Time: 16:21)



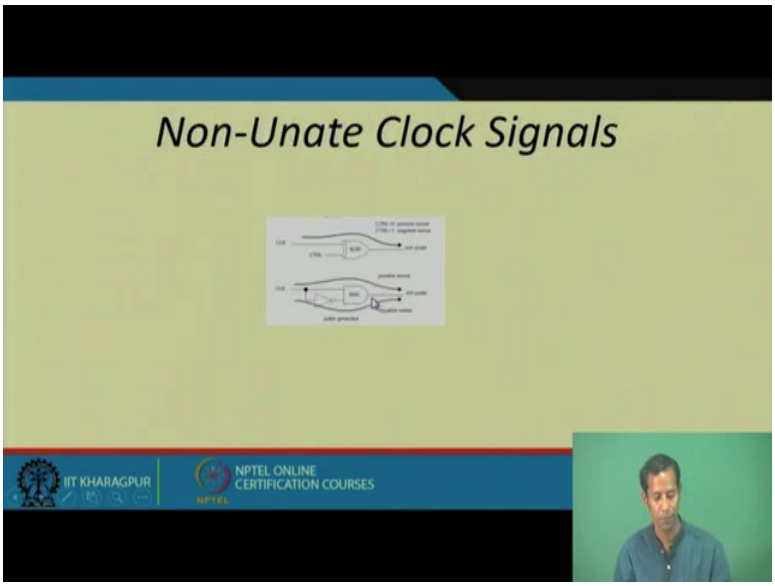
Ok and now the third; actually the thing is that the clock signals they might be passing through actually in clock tree synthesis you can use, the either the buffers or you can use the inverters ok. So, if you use two back to back inverters. So, at that time they will be just act as a buffer too. So, whenever this clock signal passes through the buffers it is the positive unate clock.

If the clock signal passes through the inverter at that time it is known as negative unate clock. If the clock again passes through two inverters back to back; that means, in series they will be known as positive unate because, that at that time it will be just act as a buffers only.

Now as we are having this clock with the enable signals ANDed in a gate and it goes to the clock of the all the corresponding sequential elements to; that means, when this enable is high at that time clock signal will pass otherwise if the enable signal is low at that time the clock signal will be 0.

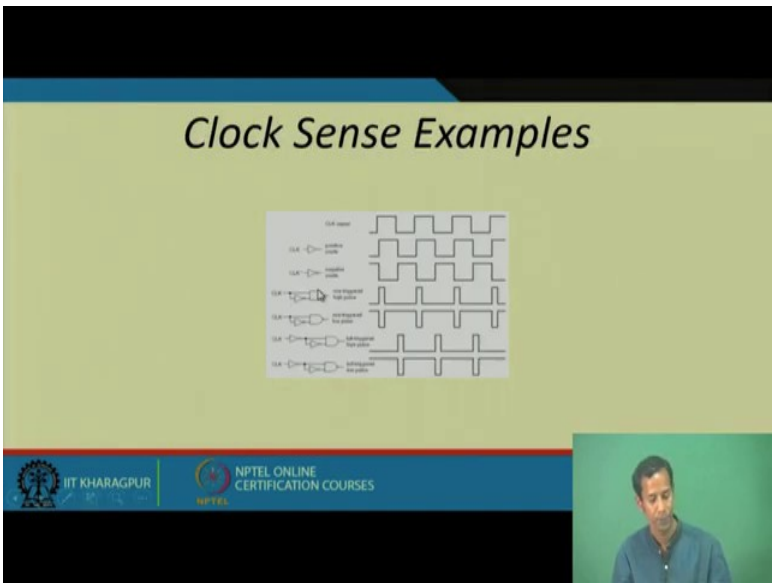
So; that means, this is if this particular clock source is used as the gated clock for low power design aspects and this is known as this positive unate.

(Refer Slide Time: 17:55)



So, then what are these non-unate clock signals. If I have this clock is basically XOR with some of the other signals, like here in this particular case clock is XOR with control in this case clock and then the inverted clock both are here ANDed. So, this is positive sense this is negative sense and then they this particular clocks these are considered as a non-unate clock signals ok.

(Refer Slide Time: 18:28)



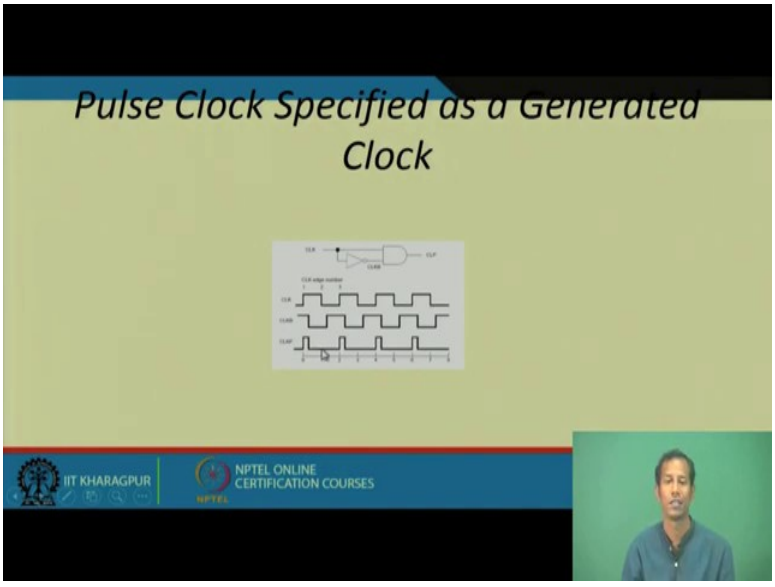
So, now clock sense example so; that means, suppose this is the clock whenever it passes through the buffers. So, it will be just delayed by the corresponding buffer delay, but the

actual that response or the waveform of the clock that will not be distorted. So, this is positive unate and this one is just if it is passes through one inverter then that is nothing, but the negative unate, then if you have connected this clock directly and another path it is just by passing through the inverter. So, at that time you will get that this edge triggered high pulse ok.

So, edge triggered high pulse means what? Rise, depending on the delay of this you will get only the pulse ok. In all other case it will be just 0. Why? Because, whenever it finds that this is the clock one and the delay of this inverter this will make the corresponding clock pulse here in this particular circuit. So, if you instead of and if you put NAND then it will be just inverted. So, here you are getting high here you will get low pulse.

So, if you put that inverter and then the corresponding this particular configuration then you will get the delay that; that means, this is the fall time; that means, here it will create this at the rise time it will create the pulse here at the time of fall it will create the pulse depending on this configuration. So, if you need to generate the pulse using the clock at that time you can use this kind of configuration ok.

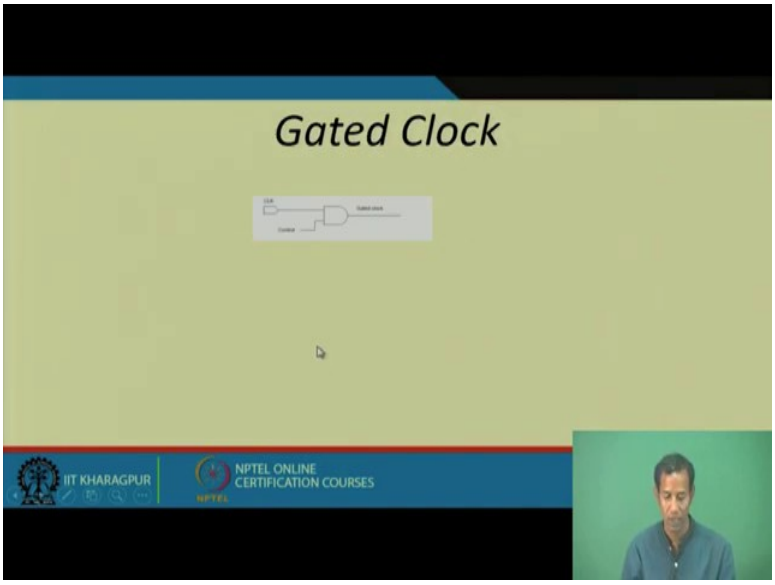
(Refer Slide Time: 20:37)



So, now we have already we have already discussed that this how this clock can be used as a pulse generator. So, the pulse width will be depending on the delay of this particular inverter. So, if you need the delay should be more at that time you have to make this particular path that means, the delay of this particular path more. That means, if it is 1

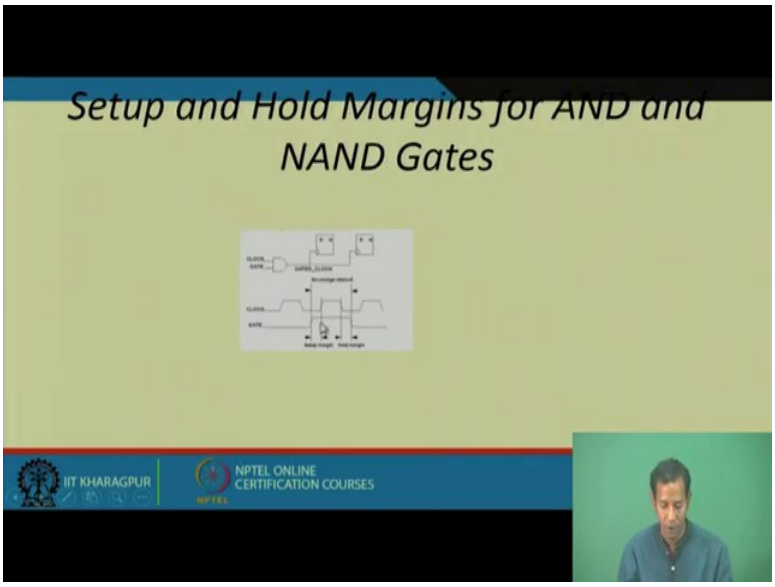
nanosecond, if a of delay of this clock; that means, this inverter is 1 nanosecond. So, the pulse width will be of 1 nanosecond if I if I put that 2 inverter or 3 inverter or 4 inverter over here. Sorry 1 or 2 3 inverter over here. So, at that time the pulse width will be 3 ok.

(Refer Slide Time: 21:33)



So, this is the gated clock, which we have already seen.

(Refer Slide Time: 21:35)



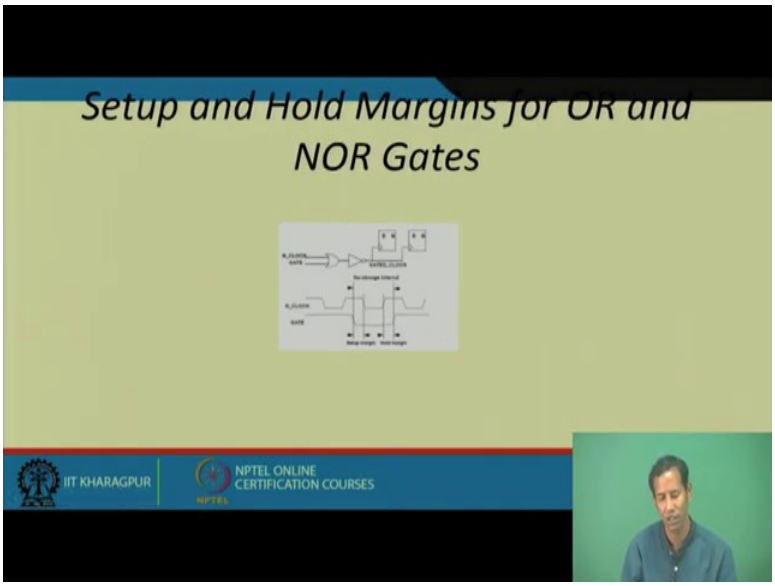
So, in the gated clock how this setup and hold margin is being considered. So, if you see that using one and gate this clock and the enable is basically ANDed. So, this is the clock and if this is the corresponding gate signals ok, if this is the gate signals. So, at that time

whenever this gate signals is getting the 0 to. So, gate signals is 0 means in AND gate one of the signals is 0 means all the time it should be the output should be 0 ok.

So, whenever this goes from 0 to 1 at that time it will find the values of clock ok. So, it whenever it is going from 0 to 1 it will find the corresponding values of the clock. And what is the data it will check what is the setup margin at this particular edge, and whenever it is going down from 1 to 0 at that time it will check for the hold margin whenever this. Again with the data is basically going down means again 0 means at that time it will not the clock will be in the 0; that means, the it will not this and gate will not pass the clock signals to this particular registers.

So, at that time it will check the corresponding hold margin and in between of this when this gated clock is 1 and the clock signal is also 1 at that time it will be no change interval ok.

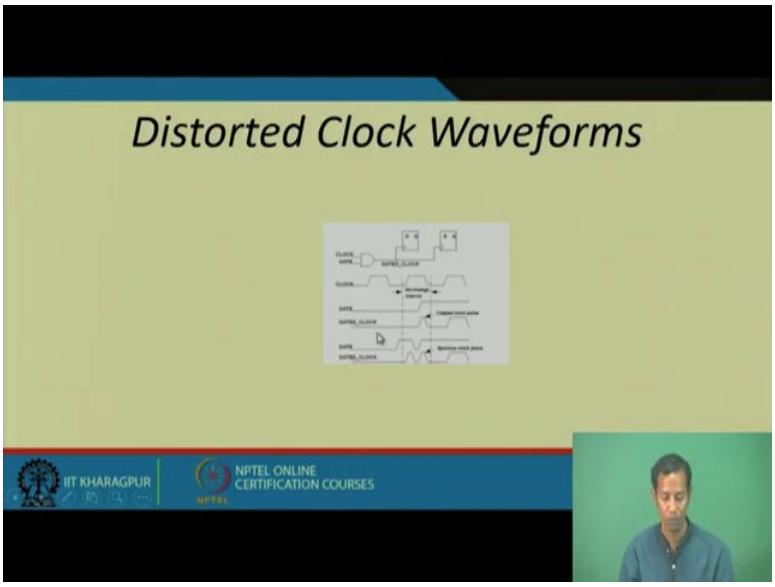
(Refer Slide Time: 23:20)



And then again if we use this OR/NOR gate type of configuration to use this for a gated clock. So, here actually gated clock means it is not that only all the time we will use AND gate, we can use this NAND gate, we can use OR gate, we can use any other gates depending on your logic of; that means, enabling or disabling the clock depending on that you can place the corresponding gates.

So, in case of NOR gate whenever that is 1. So, in case of nor gate 1 means the other input it depends on the other input. So, whenever this is this is 1 means it is 0. So, it will check what is the corresponding values; that means, what is the corresponding values in this particular clock source then again it will say; that means, when it is going from 1 to 0 it will check for this set up margin from 0 to 1; it will check for the hold margin values ok.

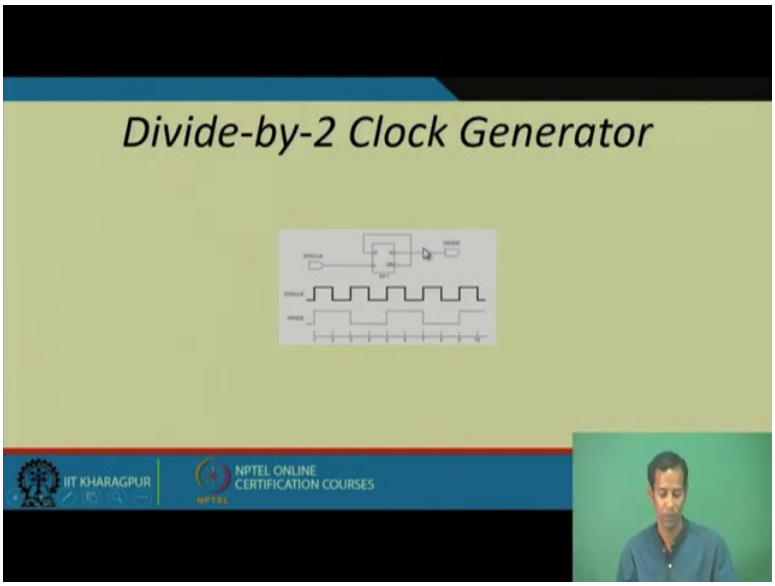
(Refer Slide Time: 24:30)



And sometimes in the gated clock design we will get the corresponding glitch ok. So, why when we will we will get the glitch as these particular signals are also having the delay; that means, this gates are already having the delay.

So, sometimes we can we will consider that later not in this particular example we have that example later we will discuss on that. In some case whenever we are having that corresponding considering the delay over here for this particular gate it may happen that you will get some glitch and the circuit response will be wrong in nature.

(Refer Slide Time: 25: 24)



Then we can actually we use this different configuration for generating the pulse through the clock source. Now, we can also use these clocks; that means, use of the clock we can make that divide by 2 clock. So, how we can make the divide by 2 clock? Suppose, this is the system clock which is going to one of the particular register cell or the flip flop cell then the flip flops Q out output if I again connected to the in the feedback if I connected to the D input.

So, at that time it will from the Q output I will get the divided by two clock. How? Because, at that time this is the system clock the output of this divide will be something like this. Means what at this particular edge it will find that this Q 1 was 1 ok. So, it will continue and it will up to this again it will find here it will find that this Q 1 is 1 and Q bar is 0. So, again it will come down to 0 then again it will check whether at this particular edge. So, this is 0 so; that means, Q is 0 means Q bar is 1.

So, again it will go on. So, something like this so; that means, here if this is also; that means, the with duty cycles of 50 percent. So, we have generate from this Q we have generate the another clock pulse which is basically the period is 2 times of the system clock. So, that is nothing, but the divide by 2 clock generator ok. So, now, using this simple logic, you can generate the divide by 4, divide by 8, divide by 16. So, any divide by this in the power of 2 you can easily make the corresponding clock generator circuit using this kind of configuration ok.

So, it is not only that we need this divide by 2 sometimes we need divide by that which is not in the power of 2. Divide by 3, divide by 5, divide by 7, divide by 9. So, at that time how we can design that that also we will consider that type of example we will consider in the next one next class.

Thank you for today's class.