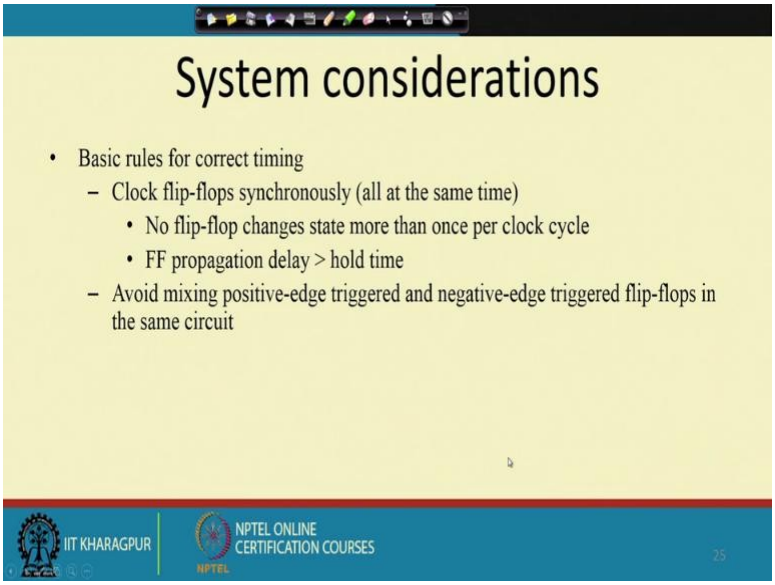
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 46
Issues in Timing Closure (Contd.)

Hello everyone, welcome back to the course on architectural design of ICS. So, we are basically discussing about the issues, which are about to happening digital circuit design ok. So, we do timing analysis for that to find out this several kind of things, like false path, then cross skew, then clock cheater. So, we will have to do. Apart from that then there are like multi cycle paths are also available.

So, many things are there which are related to the; that means, all the things we do or all; that means, whatever timing analysis we do that is to guarantees that the probability of circuit failure will be minimum 1 ok.

(Refer Slide Time: 01:08)



So, keeping that thing in mind we do this timing analysis right. So, again today we will see that, some of the things which are not yet covered in the timing analysis. So, that how we can; that means correct the timing correct the timing means: suppose. Actually till now we have discussed suppose the skew which is for a synchronous design.

So, what is synchronous design? And what is asynchronous design? So that we will see and how, we can manage this or what is the tips for using? When I used synchronous circuits or when I use the asynchronous circuits. So, that we will see if you do not use that so, at the time what happens that is also we will see in today's class.

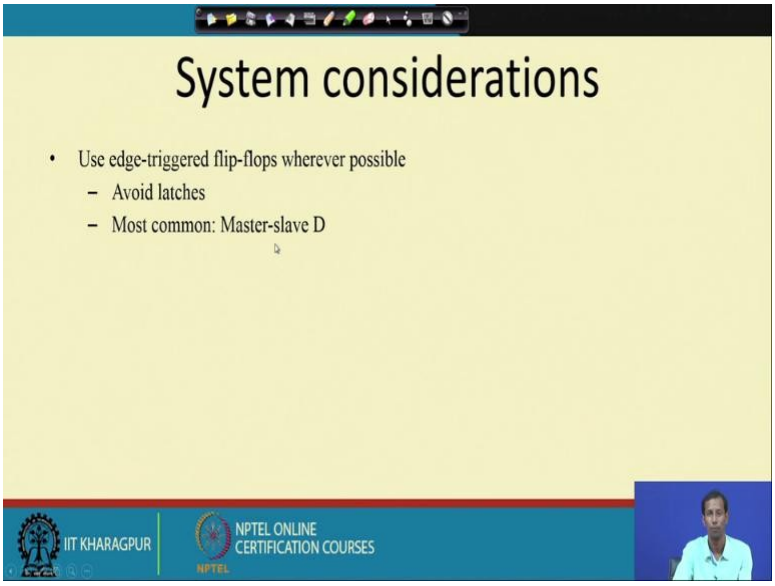
So, the basic rules for timing correcting the timing is that, the flip flop should be clock sequentially sorry synchronously. So that means, all the sequential elements what I said in the beginning, I said that synchronous circuits has more advantages ok.

So; that means, as all the flip flops are driven by the same clock; that means, all are synchronous data change is happened at the edge of the every clock edge occurs of a single clock. So, that is why here you see no flip flop changes state more than once per clock cycle. Because everyone is depended on the same clock frequency or running at the same clock frequency. And flip flop propagation delay is more than the hold time delay.

Apart from that in the same design, you never use or you not never use you just try to avoid the mixing of the positive edge triggered as well as negative edge triggered flip flops ok. Means what suppose you have single clock, but sometimes you have used positive edge triggered flip flops sometimes you have used negative edge triggered flip flop. For a single design consideration of these two facts is not beneficial or it basically creates the problem.

Why creates the problem: because at the time at synchronization or that means, the changes of the data you may get problem at that time ok.

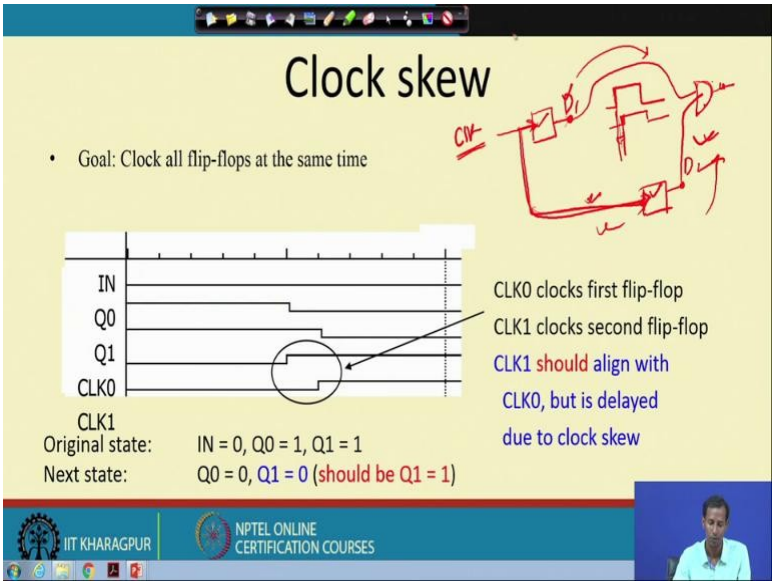
(Refer Slide Time: 03:40)



So, then use of this edge triggered flip flops instead of latches, why; because latches level sensitive not egde sensitive ok. So level sensitive means, it is not that it depends on the only on the clock and then; that means, the highest priority at that particular circuit is clock.

So, level sensitive means it depends on the clock as well as the data at the same time ok. No one is getting the highest priority among them. So, that is why always try to use edge triggered flip flop instead of latches, and always try to use this Master Slave D flip flop architectures.

(Refer Slide Time: 04:37)



So, this is the system consideration which we need to be keep in mind, whenever you are designing that your system.

So, then what is this clock skew? Clock skew is what I said the clock skew is the; that means, whenever I am connecting two different flip flops. So, at that time it is not that every time, this every clock is getting the; that means, the clock edge is basically occurring at the same time, why it is not occurring at the same time, because of the propagation delay which is associated for the clock pin to that particular element; means what clock skew suppose, I am having the flip flop over here and another flip flop over here ok.

So, this is connected through clock, suppose if this is the clock. So, the clock which is coming over here and the clock which is coming over here, they have a time difference why they have a time difference, because this is a longer path to be traversing need to be traverse for the clock signal.

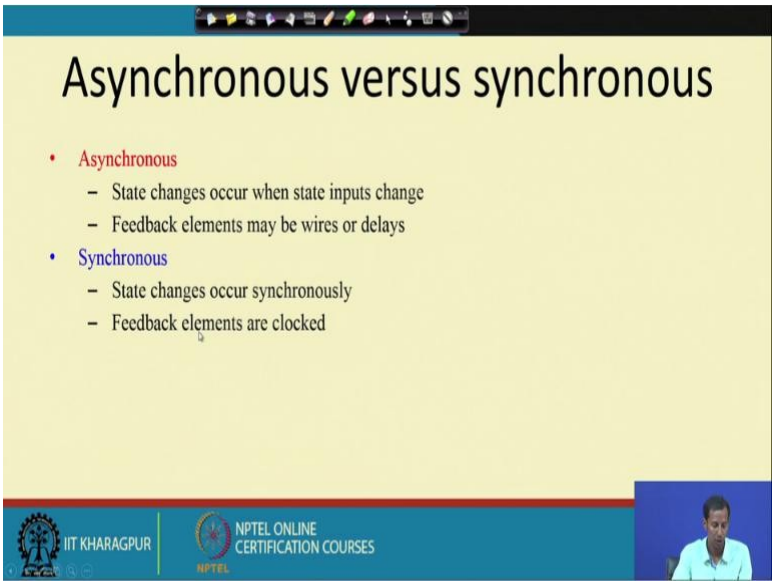
So this will, itself this path will containing some of the edge delayed. So that means, this clock will be delayed. So, the if the edge at this particular time if this edge occurs at this will not; that case it will occurs after some of the time, at this particular point after sometime.

So, this is basically the skew clock skew and when it is beneficial and when it is again starts that we have already seen all right, but the thing is that based on as this is not; that means, the edge is not proper for this and this. That means, the change of the data at these two particular point there will be change. That means the change of the data that will occur at different time for this D 1 and D 2.

So, which is not means what if I this? This two data is changing at different time, now again suppose this D 1 and D 2 is connected to some of the logic let us consider. So, at the time I will get error why I will getting error? Because this is coming at different time, this is coming at different time.

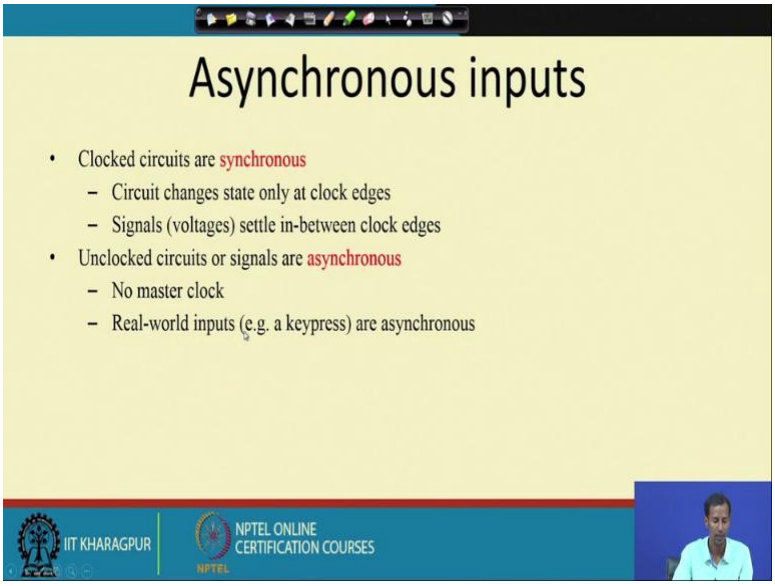
So, that is why I can get erroneous result over here. So, for that reason it is; that means, advisable that you have to make that clock skew to 0. So, clock skew to 0 means the time requirement for this and the time requirement for this that become that remains same ok. So, that is the concept of clock skew.

(Refer Slide Time: 08:15)



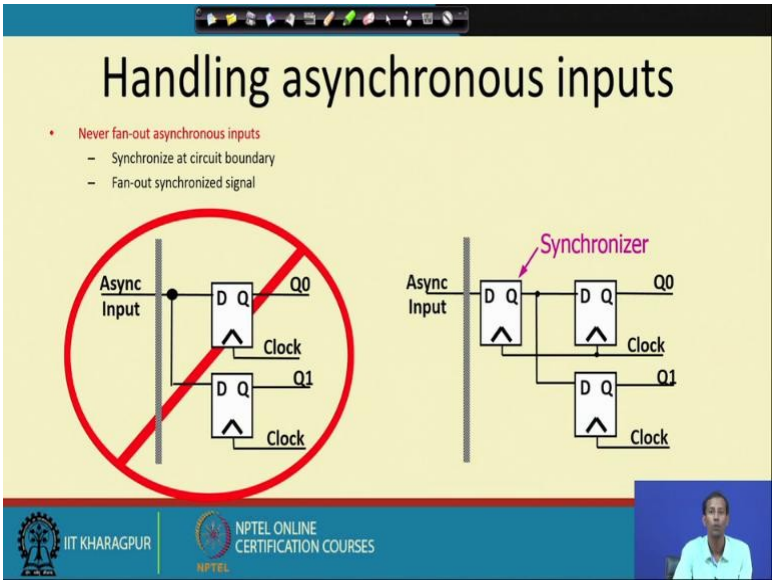
So, then what is synchronous design and what is asynchronous design? In synchronous circuits, the state changes occur when state inputs changes. The feedback elements maybe wires or delays in asynchronous circuits, but in synchronous circuits the state changes occur synchronously feedback elements are all clocked ok.

(Refer Slide Time: 08:42)



So, what is that if we take the example? So, at that time it will be much more visible to you. So, asynchronous circuits means what. The circuit changes state only at clock edges we will take the example of this.

(Refer Slide Time: 09:07)



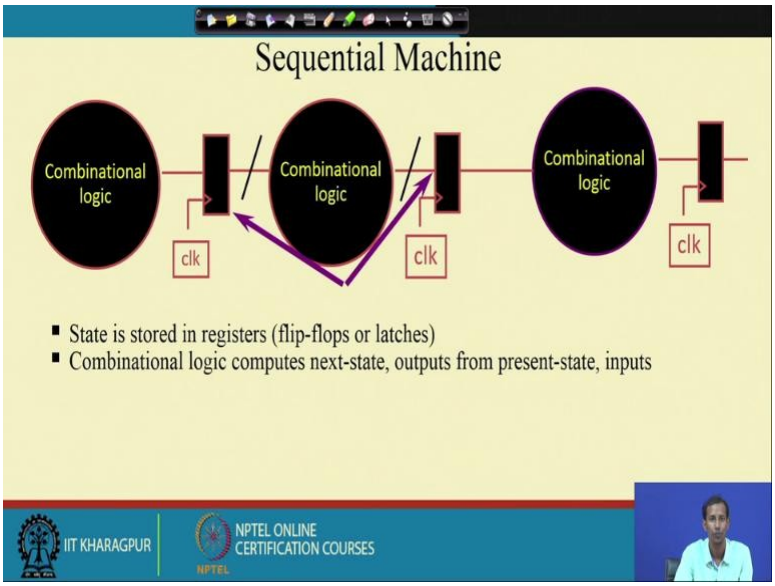
So, suppose I am having 1 asynchronous circuit something like this ok. Now asynchronous circuits asynchronous input like this means if I just connected it two different D flip flop which is clock at different end. So, at the time asynchronous input means it is coming at different time. So; that means, at the time it will reach to D 1 at

different time D 2 at different time. So, that is why it is not; that means, for the asynchronous input, asynchronous input means if the input is not arriving to the flip flop at the proper time.

So, at that time what we have to use? We have to use 1 synchronizer. So, synchronizer means what you are you have to put another D flip flop ok, another D flip flop and then again that particular output will be connected to the other two flip flop, which are connected over here. That means, suppose one particular signal input, if it is connected to several D flip flop over here. So, at the time the first thing is that you have to make that signal that signal has to be passes through the registers, then you connect to each of the resistor element, which is connected over here ok.

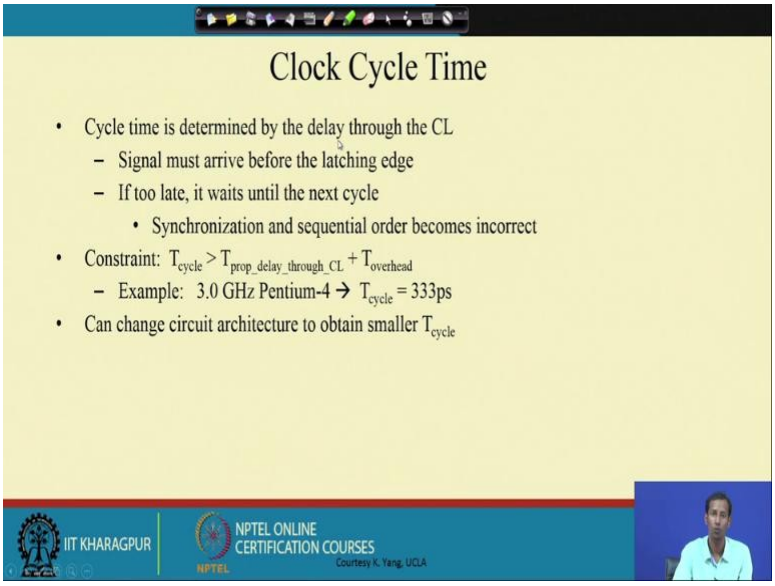
So, this particular technique is known as synchronizer. Otherwise what will happen you may get some of the setup time or hold time violation ok.

(Refer Slide Time: 10:51)



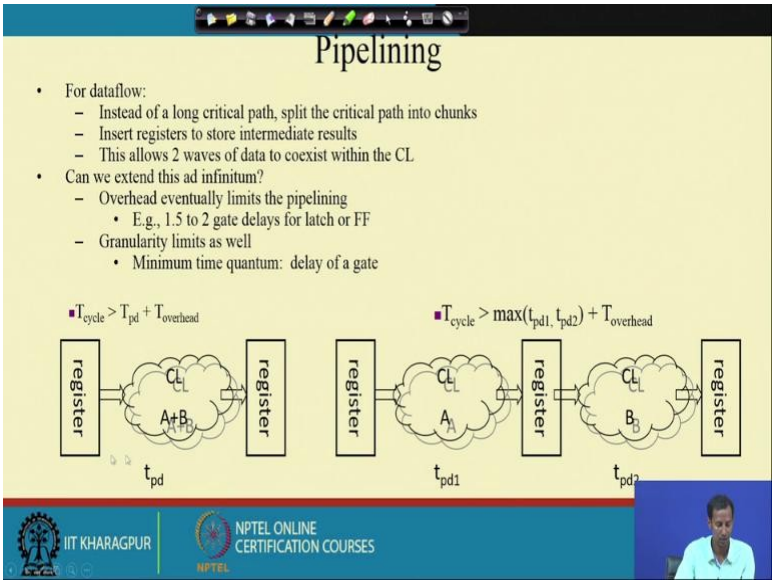
So, that we will see now; that means, now we will see those types of things where this kind of that means violation can happen ok.

(Refer Slide Time: 10:59)



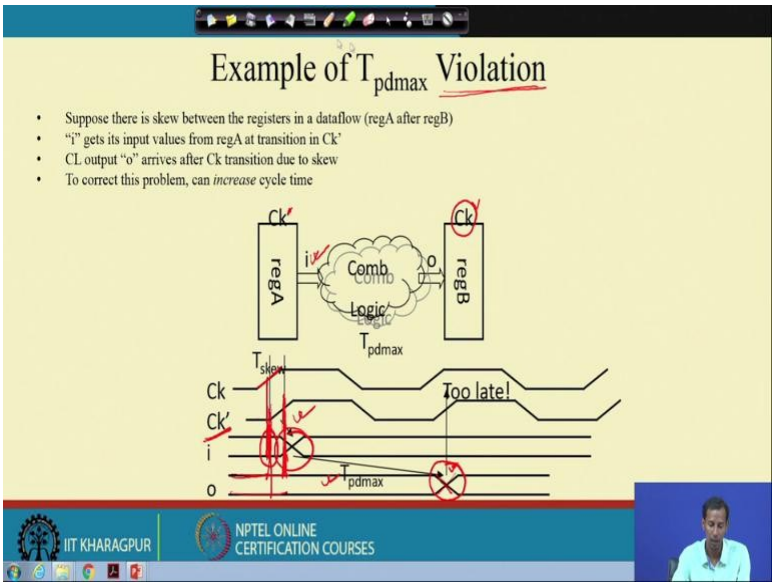
So, that means, the same thing we have already considered suppose how to or what is the; that means, need for this competition of this clock cycle time? Because this clock cycle time is determined by the delay through this clock cycle the signal must be arrived before the latching on edge, and if it is too late it waits until the next cycle.

(Refer Slide Time: 11:31)



So, now this is not the case suppose this is one combinatorial block and then we have to register. So, we this we have already seen in the previous example.

(Refer Slide Time: 11:52)



So now, if I consider this about this violation when I can get the violation ok, so suppose I am having this combinatorial circle something like this and then I am having 2 registers over here right.

So, now I am having the clock, this is my clock and then if I am having the skew clock skew. So, at the time this clock will be delayed right. So, if this i is the input to the combinatorial circuit then, because of this skew. Suppose at these particular point at these particular point, whenever this clock is basically changing the data. So, at these particular edge this is basically finding the corresponding positive edge and what is the value over here i? So, the output is 1 over here or the output should be changed over here ok.

So, if I just take for this clock whenever I am getting the positive edge. So, at the time it will try to find out the corresponding data over here ok. Now at this particular edge as this is finding there is these values are basically changing not changing. So, it will try to hold it value, then again what is happening for this? It is this is clock bar this is clock bar.

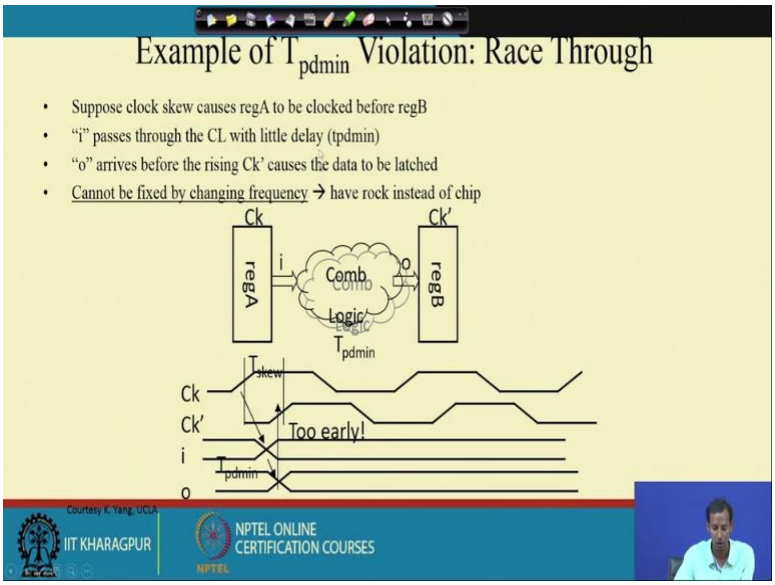
At these clock bar what is happening? At these particular points, this particular point as these data is changing. So, again it will try to change it is value, means what? Because of this skew the proper change in the data that is basically missed ok.

So, because of these things happening we are getting the violation ok. So, here you see suppose there is skew between the registers in a dataflow, which is register A and register B i gets the input values from register A and transition at CK dash then the clock output o

arrives at clock transition due to skew, to correct this problem can increase the cycle time ok.

So, that is why if I; that means, consider the corresponding i; that means, at these particular point as these i is basically changing it is data because of this skew that you just consider as these 50 percent no change in the data, but because of this skew data is changing at these particular point. So, data is changing at these particular point means it will create the problem to the output of this, which is basically creates the violation ok.

(Refer Slide Time: 15:19)

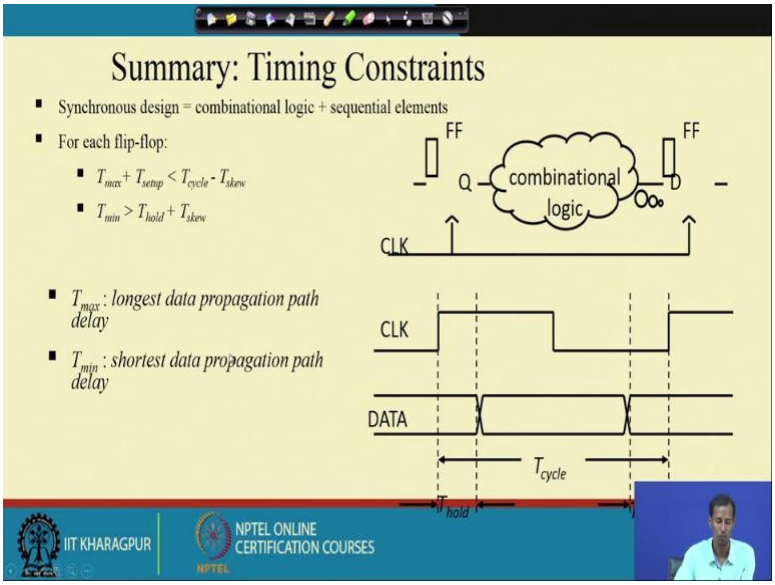


So, though because of this skew I we can get at sometime we can get this kind of thing ok. So, then again another example of this T pd minimum violation, which is the race through. Suppose the same kind of structure if you just see and here because of this clock skew, because of this clock skew the data changing at i that can happen early.

Here what is happening? Here the skew because of this skew the data is basically changing lately, but here you see because of the; that means, there is a few change in the; that means, in the clock to clock bar. So, the data is basically changing very early. So, this is another T pd minimum violation, this is T pd maximum violation.

That here you see here this input is basically depended on the clock and this register B is depended on the clock bar ok.

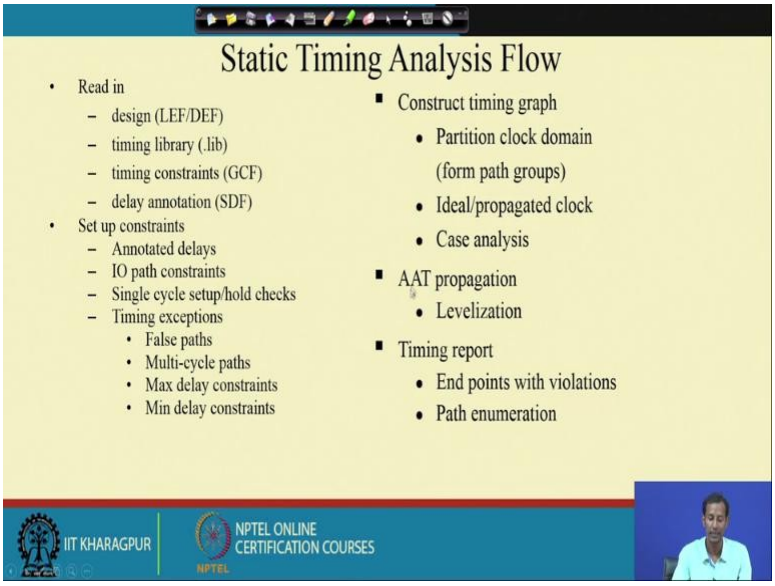
(Refer Slide Time: 16:29)



So, based on that you can; these timing constraints that can be summarised as follows for any synchronous design ok. It will be combinational logic plus the delay which are associated with the sequential elements ok. For each of the flip flop this condition has to be satisfied that T max plus T setup that will be lesser than the T cycle minimum minus T skew and the T minimum will be greater than the T hold plus T skew. Where T max is the longest data propagation path delay and T minimum is the shortest data propagation path delay ok.

So, these are the other things.

(Refer Slide Time: 17:37)

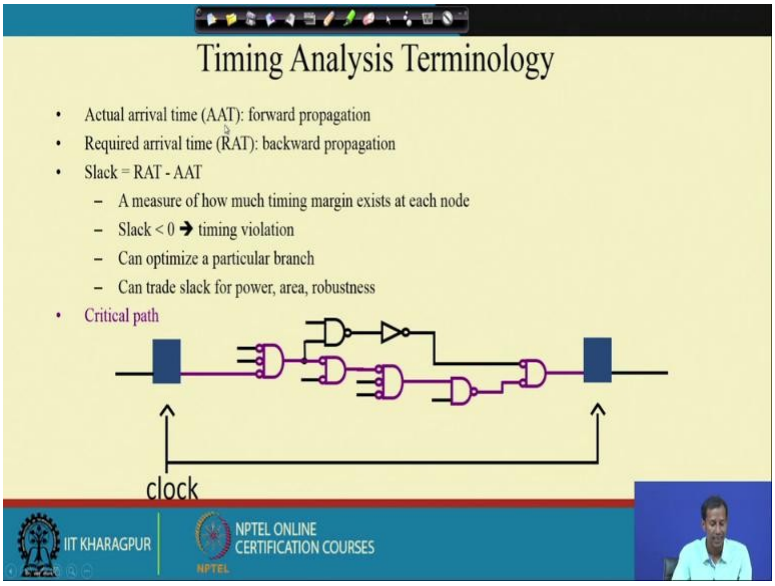


So, in static timing analysis flow what we require basically? This is very much important why what we require? Whenever we are doing static timing analysis at the time, we need to design these or we need to; that means, consider these read in the design, then the timing library, then the timing constraints, then the delay annotation. That means, all these information if I provide to the EDA tool then only that tool can give you the timing analysis report along with that whenever we are doing this timing analysis.

So, at the time you have to follow this annotated delays then IO path constraints, then single cycle setup or hold time checks. Then timing exceptions for false path, then multi cycle paths maximum delay constraints and minimum delay constraints. So, in the false path consideration what we have seen that, we have to mention the tool that this is the false path we have to show from the beginning to the tool right. So, here also that is why whenever we are doing; that means, setting up the constraint file. So, at the time we have to set the timing exceptions that, this is the false path.

So, you set the timing exceptions for this path or you check whether there is any setup time violation or hold time violation setup time, and whole time violation can happen, because of this skew or because of this glitter this can happens. And then construct the timing graph based on this partition clock domain or ideal or propagated clock or case analysis.

(Refer Slide Time: 19:31)

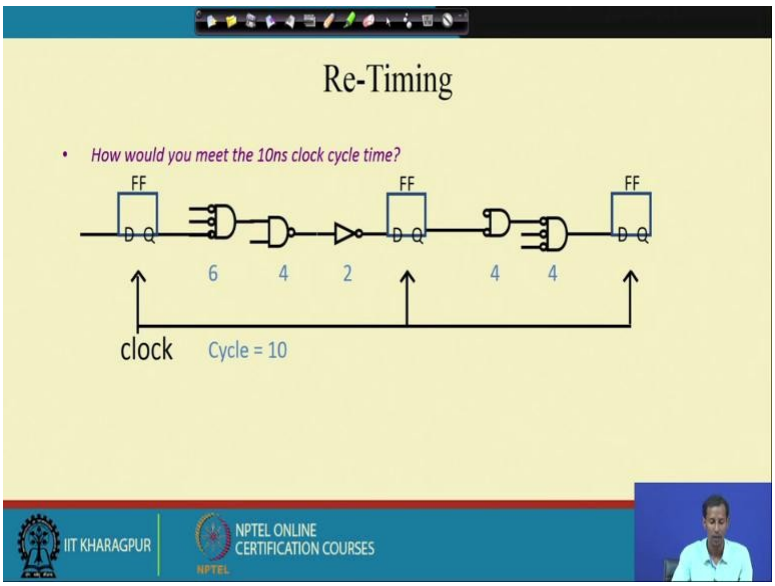


And then again this early arrival you have to; that means, for this slack calculation you have to calculate this actual arrival time as well as required arrival time.

So, based on that you can calculate this slack ok, where this slack is very much important to get or to optimise the corresponding clock period ok. So, these are the important factors which are which has to be consider whenever we are doing this static timing analysis right. So, then this is the false path example.

False path example we have already seen.

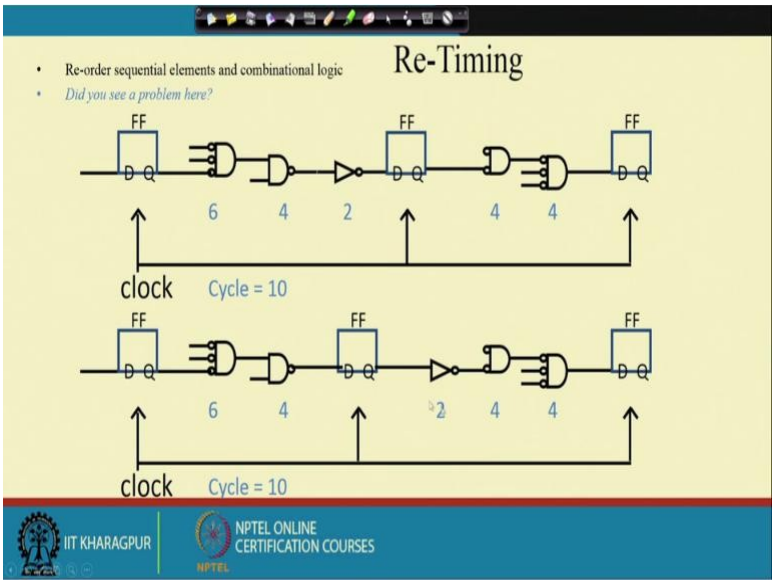
(Refer Slide Time: 20:09)



Now, another thing suppose here we are having 1 circuit right. So it says that, how would you meet the 10 nanoseconds clock cycle time for this particular path? That means, when these clock cycle will be 10 nanoseconds.

So, as you see in this particular path. So, this is register to register, there is one combinatorial circuits which is taking 6 plus 4 plus 2 means 12 nanoseconds over here and this particular path is taking 4 plus 4 is 8. So, clock period of 10 is it basically can I do this clock cycle period as 10. If I do the clock cycle period to 10 so, at the time as these time requirement is 12. So, the circuit will fail to work.

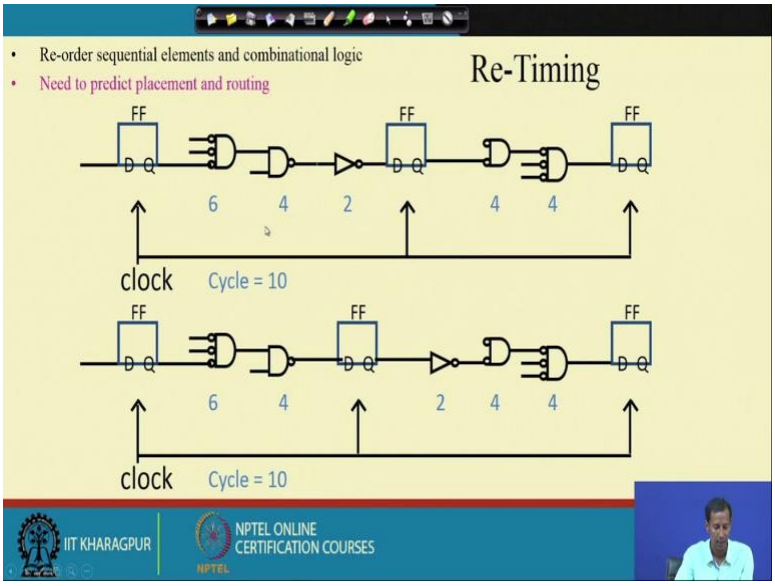
(Refer Slide Time: 21:19)



So, at the time what I have to do? I have to do a technique which is; that means, named as this re timing. So, what we see that here in this path it is taking 6 plus 4 plus 2; that means, 12 over here and in this it is having 4 plus 4.

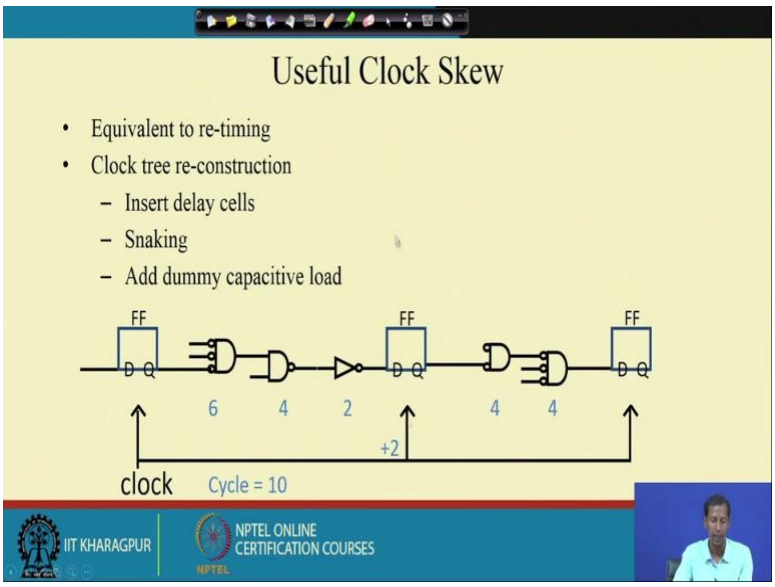
So, if I place this flip flop in these particular position, so at the time this position now it will take 6 plus 4 10 and here 2 plus 4 plus 4 that is 10 ok. So, you see the flip flop has been just replace to this particular position to make the balanced in this 2 path 10 over here.

(Refer Slide Time: 22:15)



So, 10 over here now we can do or I can; that means, said that cycle time as 10. But; that means, is there any problem is there any problem? No, because after doing this after doing this, we can as we have made this; sorry, we have made as this two particular structure is balancing in both the direction.

(Refer Slide Time: 22:55)



So, now, this will be this both that means, things or I can set the clock cycle to 10 nanoseconds ok. So, another thing also what we can do; we can add the skew of 2. So, skew of 2 means if I put clock in the same direction to the data. So, at the time what will

happen? It will just reduce the clock period by 2; that means, whatever is the delay. So, that will be minus if I put the clock on the same direction with the data. That means, at the time delay will be minus skew amount will be minus, means what at that time it will be 6 plus 4 plus 2 minus 2 so; that means, it will become 10.

So, at the time easily set the clock cycle tool 10. So, that is another way of making this clock cycle period to 10 so; that means, then in the previous case, what we have done? We have use this re timing or re ordering or changing the position of the flip flop making the delay both the hand same, which is 10. In this particular case, we have used this we have intentionally added clock skew of 2, 2 unit of time to make this clock cycle of 10 unit of time ok. So, how we can do? That means, we can increase the clock skew, we can insert this delay cell or we can add some dummy capacitive load. That we can do to introduce this clock skew in this particular path ok.

So, this is a technique by which we do to avoid the violation ok. So, this is the end of the this timing closure; every time I am telling you that, it is not the things that this is the end of it, there are more on this, but it is not possible that all the things will be covered in this particular course. If you are more interested then please let me know.

With that thanking you, so this is the end of today's class.