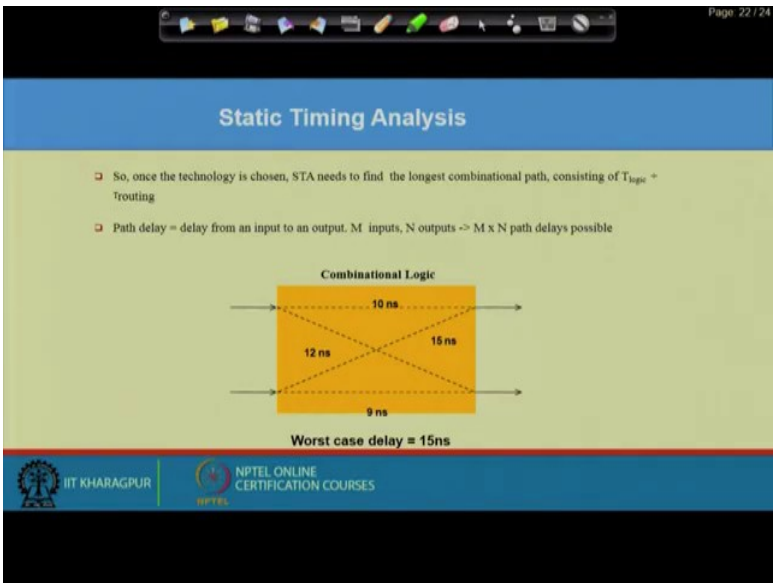
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 53
Timing Analysis Basics (Contd.)

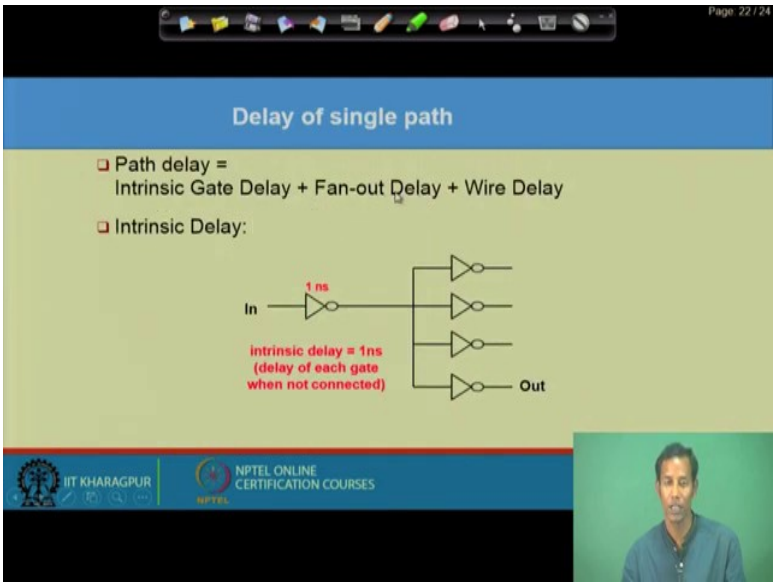
(Refer Slide Time: 00:37)



So, hello everyone, welcome to the course on Architectural Design of ICS. So, we are basically considering the Timing Analysis Basics whether that is static timing analysis or dynamic timing analysis. So, they are very much important whenever we are considering any digital circuit design or digital integrated circuit design. So, in static timing analysis we have already seen that ok, there are there may be more than one path or there may be; that means, many paths.

So, from input to the output which path is basically considering the maximum number of delay. So, that is the cycle time requirement of your circuit design ok.

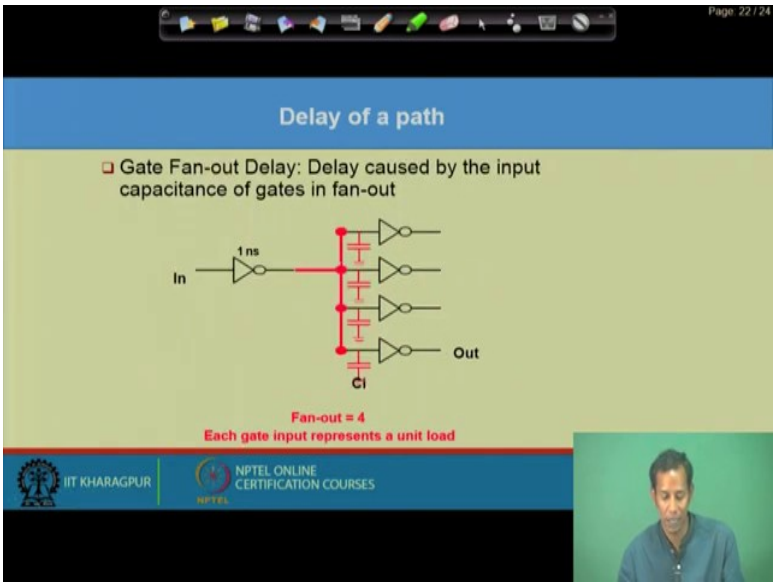
(Refer Slide Time: 00:59)



So, the delay of a single path that can; that means, that can be calculate based on the intrinsic gate delay plus this fan out delay plus the wire delay. So, this intrinsic delay ok. so, the intrinsic delay means the inherent delay which corresponds to the each of the gates and if one of this particular inverter is now driving four different other cells. So, at that time their delay will be information will be this fan out delay if it drives two; that means, two gates here.

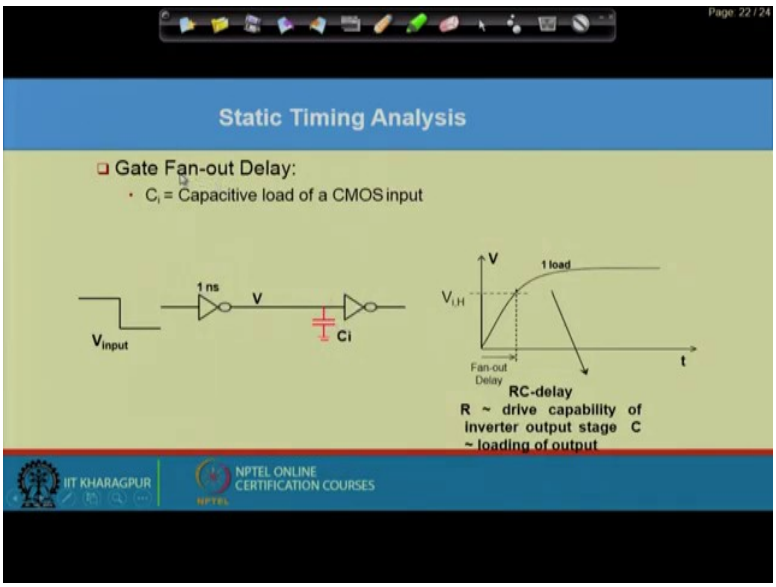
So, at that time what will be the, what will be the amount of delay for this gates; if it is driving more than; that means, if it is; that means, 6 cells here so, at that time delay will be more for this particular cells. As well as this corresponds actually there are 4; that means, paths it is driving means that there are 4 paths, there is 4 wires which is connected from this particular edge. So, the wire delay will be also considered in the corresponding path delay of the combinatorial circuit delay.

(Refer Slide Time: 02:22)



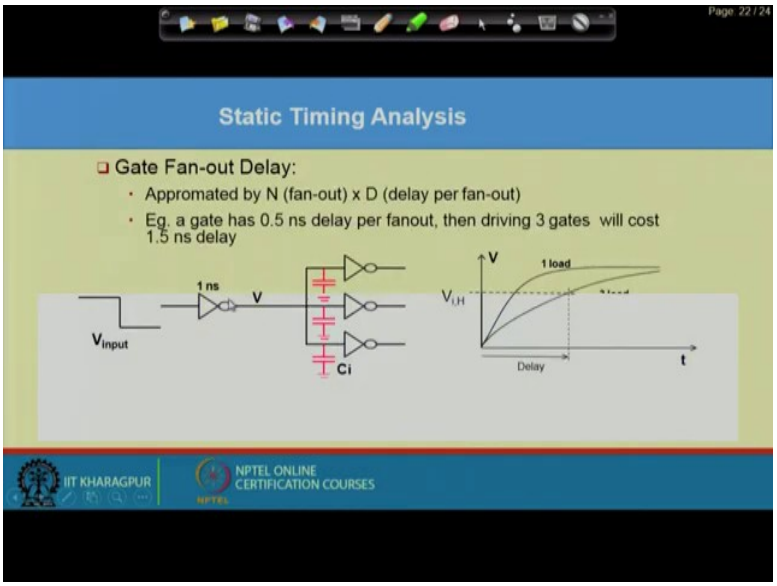
So, this as this fan out is 4. So, the gate fan out delay is caused by the input capacitance of the gates in the fan out cells ok.

(Refer Slide Time: 02:34)



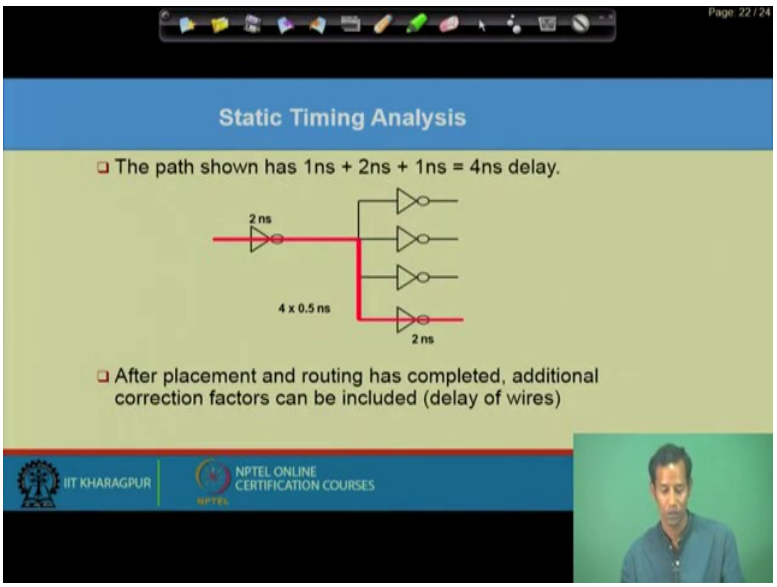
And then how these gate fan out cells basically this gate fan out delay is calculated based on the capacitive load of the CMOS inputs which it is driving ok. So, depending on this R and C value we can calculate that how much delay it corresponds to that what is the number of fan out for this particular gate ok.

(Refer Slide Time: 03:01)



So, depending on that we can calculate or we can compute or in static timing analysis tool this is being computed. So, if one gate, actually how we can compute is that this approximated by the factor N which is the fan out and then the delay; that means, D is the delay amount for per fan out ok. So, if as an example if we consider that a one gate has 0.5 nanosecond delay per fan out. So, when it is driving 3 gates will cost 1.5 nanoseconds of delay. So, this information will be basically stored or this information will be changed depending on different library ok.

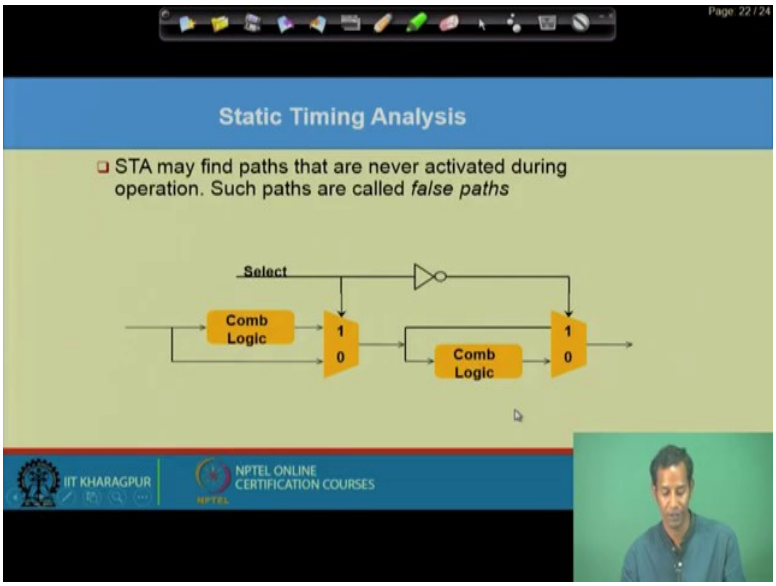
(Refer Slide Time: 03:49)



So, then in in total if we calculate the total path delay of any circuit which corresponds to the 2 nanosecond delay over here and then if that is also 1 so, the intrinsic the path delay or this the fan out delay if one of the fan out delay is let say 0.5 nanosecond. So, for 4, it will be of 2 nanoseconds delay ok.

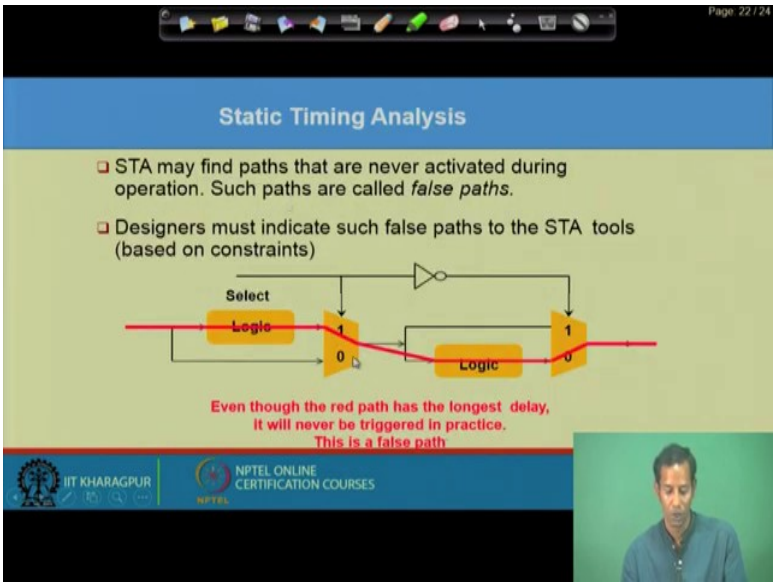
So, the path it is showing is that the 1 is the corresponding this 1 is this the delay for this particular gate and then 2 is for this fan out delay this fan out delay and this wire delay let say that is of 1 so, the total path consist of 4 nanoseconds of delay. So, after placement and routing has completed additional correction factors can be included which is the delay of wires.

(Refer Slide Time: 04:56)



So, now STA; that means, STA is nothing but the Static Timing Analysis, them they may find paths which are never activated during operations so, such paths are called as false paths. So, what is the; that means, example of false path is that; suppose we are having two paths; that means, two logic here this is the combinational logic, here we are having 1 and 0 and again we are having this combinational path logic which is having this 1 and 0.

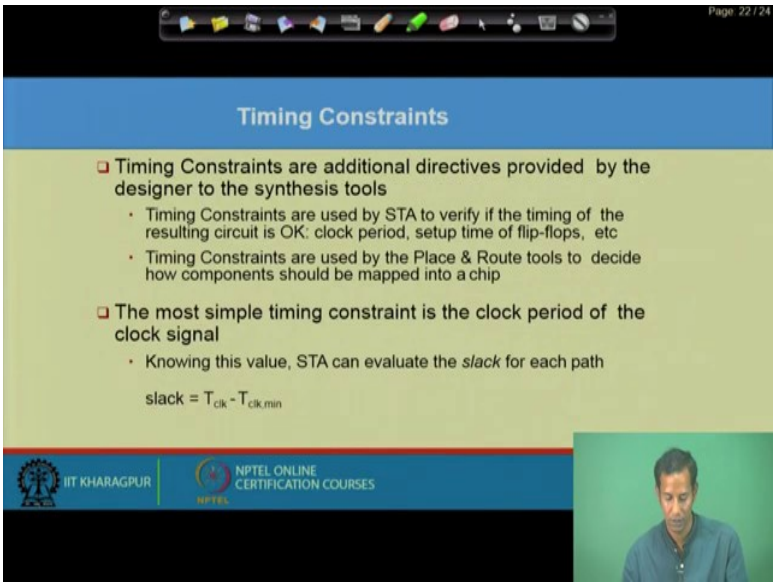
(Refer Slide Time: 05:33)



So, whenever this select equals to this select let say this select equals to 1. So, at that time particularly this particular path whenever this is 1 means this is 0. So, the corresponding logic will pass from these to this ok.

So, the STA may finds that they are that there are never even activated path during the operations and such of those paths are called as false path and the designer must indicate such false paths to the STA tools so that, it does not actually calculates or it does not include the timing information of those false paths otherwise what will happen? The timing information will be misled to the some other values. So, even though the red path has the longest delay it will never be triggered in practice because this is a false path.

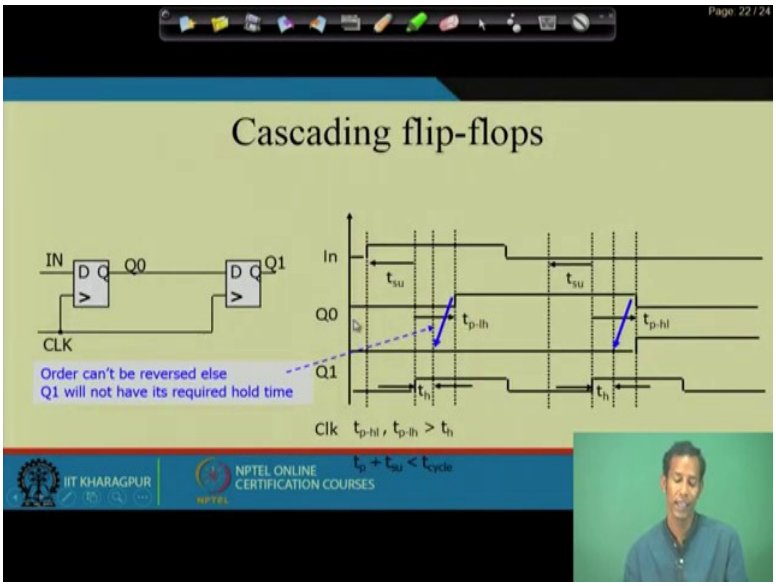
(Refer Slide Time: 06:49)



So, the timing constraint are basically additionally directives provided by the designers to the synthesis tool and this timing constrains are used by this static timing analysis to verify if the timing of the resulting circuit is ok; that means, in terms of this clock period requirement, in terms of this setup time or this hold time violations are there or not ok.

So, after this place and route we use this timing constraint also be used to the place and route tools to decide how the components should be placed into a corresponding chip area, within the chip area. Why? As we have already said that this clock skew or this delays amount, there basically calculated based on the how their connected to each of them to another one ok. So, then we have already known that what is the slack, slack is the required time minus the actual arrival time ok.

(Refer Slide Time: 07:56)

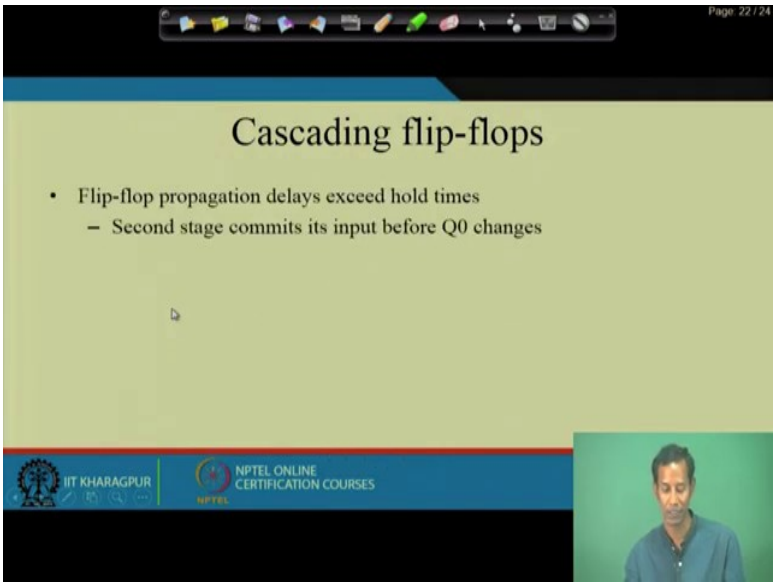


So, if we cascade these two flip flops in; that means, in series. So, at that time suppose this this clock is connected to this and this Q 0 and the Q 1 flip flop. So, this is the input which is again going to the; that means, the second the registers.

So, if we see that this is the input signals, this is the variation in the input signals and this q 0 is basically is this one and then Q 1 will be depending on the clock, depending on the clock edge ok. So, this Q 0 value is 0 sorry, this in is; that means, in is 1 so; that means, after sometimes the Q 0 value will be 1 which is this the amount of t; that means, the time which is going from t high to low or the low to high ok.

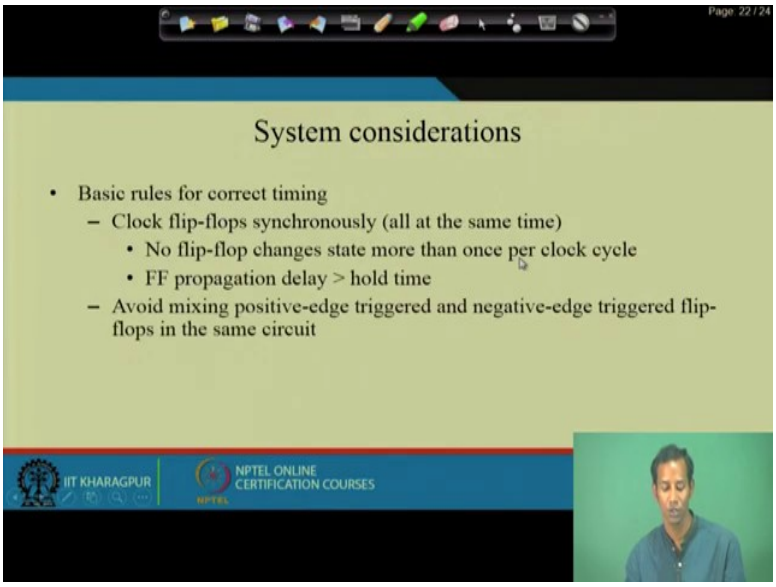
So, the t this setup plus the t requirement for this particular flip flop or the delay of the flip flop that is the minimum requirement of the t cycle over here in this particular circuit design ok.

(Refer Slide Time: 09:19)



So, whenever we are cascading two flip flop. So, at that time this the propagation delays if it exceeds the hold time at that time the second stage commits it is input before the Q 0 changes ok.

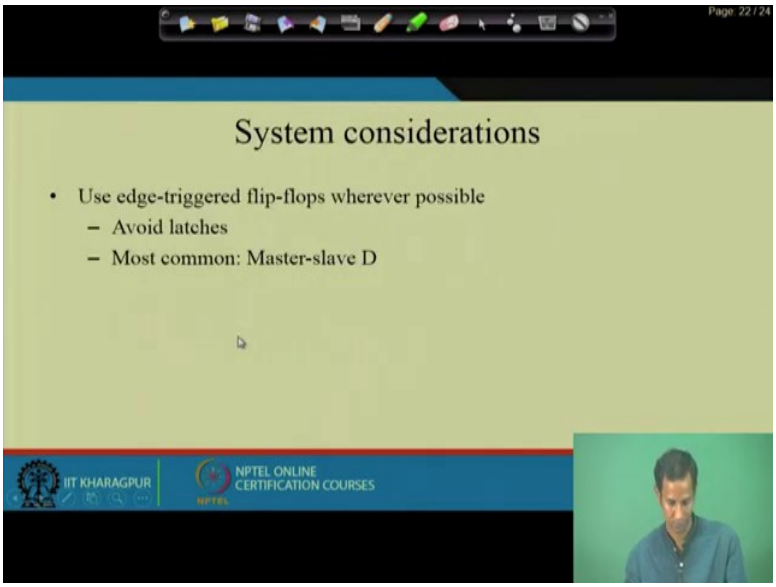
(Refer Slide Time: 09:36)



So, for that particular things we have to consider the system like how we can correct or there are some basic rules to correct the timing is that whenever that two flip flops are synchronously connected to; that means, they are they are operation or they are operating at the same event of the clock. So, no flip flop changes the state more than once per clock cycles

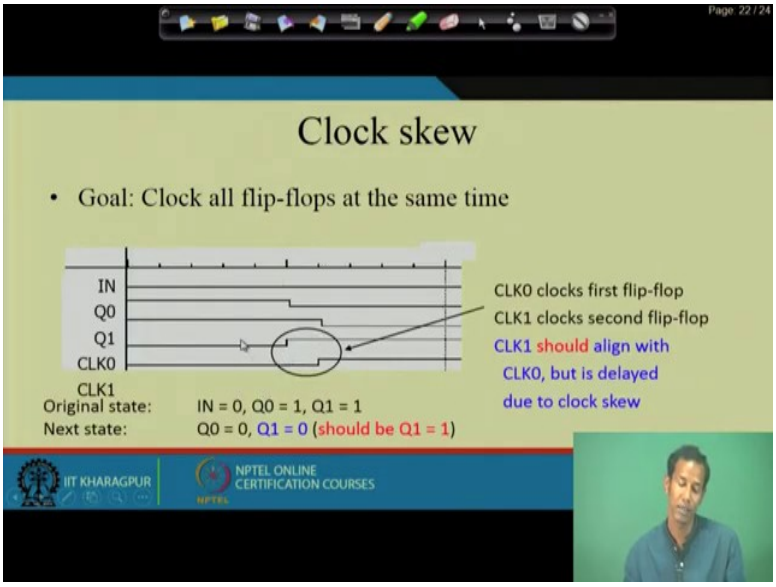
or the flip flop propagation delay should be more than the hold time. And always avoid the mixing of this positive edge triggered and the negative edge triggered flip flops in the same circuit.

(Refer Slide Time: 10:28)



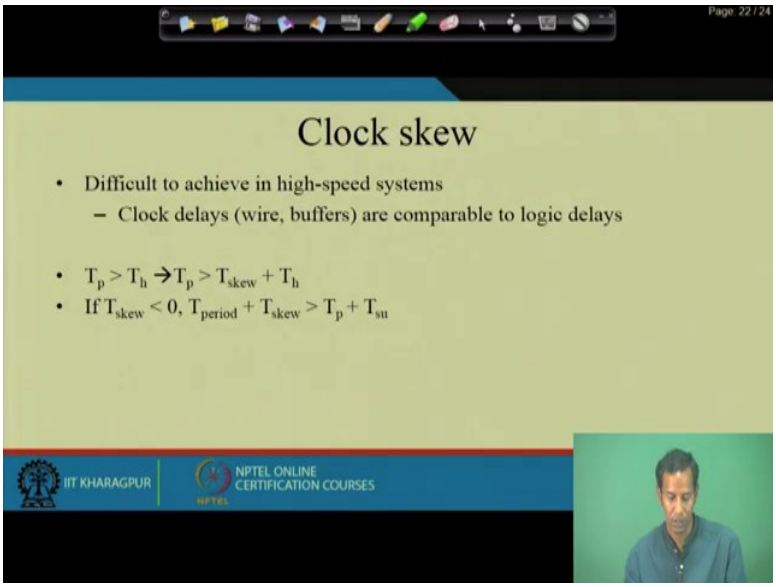
So, again when we use edge triggered flip flops where whenever it is possible always try to avoid the latches and most commonly used of the; that means, this edge triggered flip flop is the master slave configuration of the D flip flop ok.

(Refer Slide Time: 10:46)

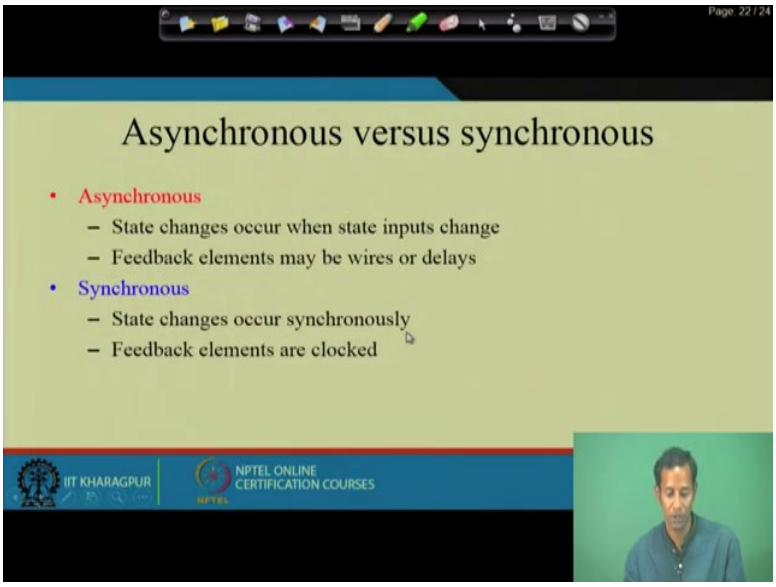


So, the clock skew is we have already discussed said; that means, if there is the depending on the arrival of the clock, if there is a time gap between the clock edge so, at that time how much is the time gap that is the clock skew.

(Refer Slide Time: 11:05)



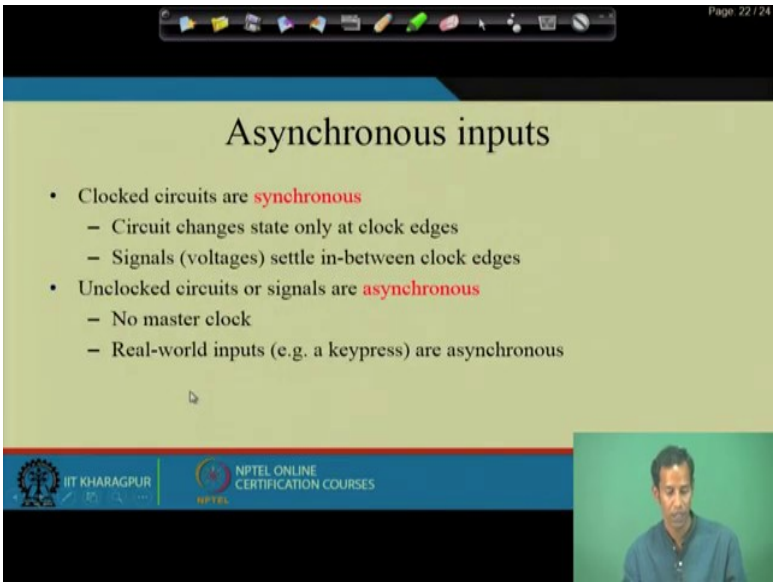
(Refer Slide Time: 11:08)



So, this the calculation of the clock skew; we have already said and then actually what is the what is a difference of the synchronous circuit and the asynchronous circuit? So, the in asynchronous circuit the state changes occurs when the state input changes. And the feedback elements may be wires or delays.

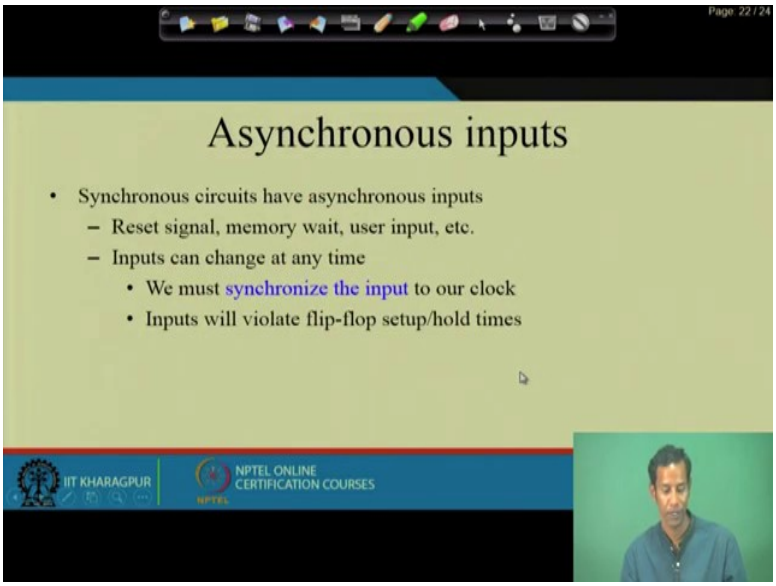
But, in synchronous circuit the state change occur synchronously and in the feedback elements are clocked; that means what in asynchronous circuit we may have different clock which has been different event or the triggering point at different time instant, but in synchronous circuit all that; that means, event occurs at the same timing instant. So, that every; that means, every this sequential element they can fire or they can; that means, start their working at the same instant ok.

(Refer Slide Time: 12:14)



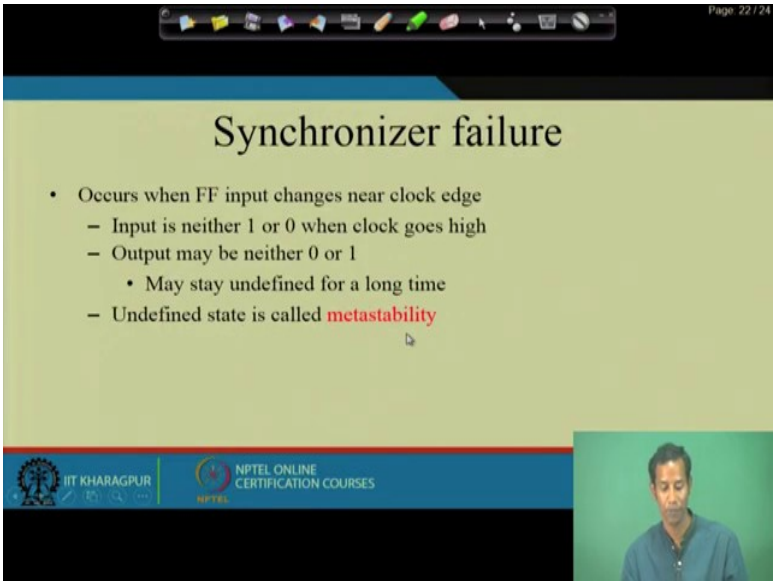
So, when actually this asynchronous input means that to deal with this asynchronous circuit that is much more critical to work with the synchronous circuit. So, when this clock circuits are synchronous; the circuits changes it is state only at the clock edge whether that may be positive edge or whether that may be at the negative edge and the signals settle in between the clock edges only. In un-clocked circuits or the signals are asynchronous so, no master clock, the real world inputs are asynchronous like this key-pressing all this real world inputs are asynchronous.

(Refer Slide Time: 13:01)



So, whenever we are dealing with these asynchronous inputs so, at that time the synchronous circuits may have this asynchronous inputs like this reset signals, memory wait, and user inputs etc. The input can change at any time so that, we must synchronize the input to our clock or with our clock and inputs will violate the flip flop setup and hold times ok.

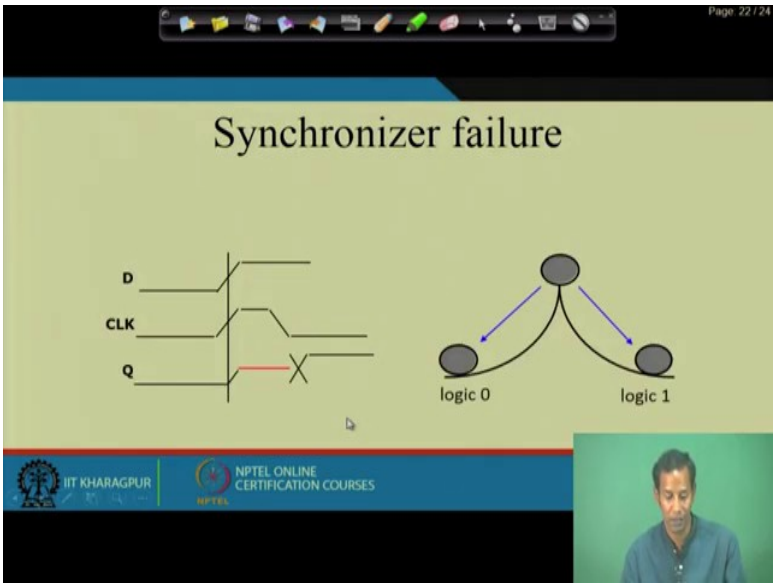
(Refer Slide Time: 13:30)



So, with causes this synchronizer failure. So, when it occurs? It occurs when the flip flop input changes near to the clock edge. So, what will happen? At that time the input is neither 1 or 0 when the clock goes high, the output may be neither 0 or 1, they may actually the they

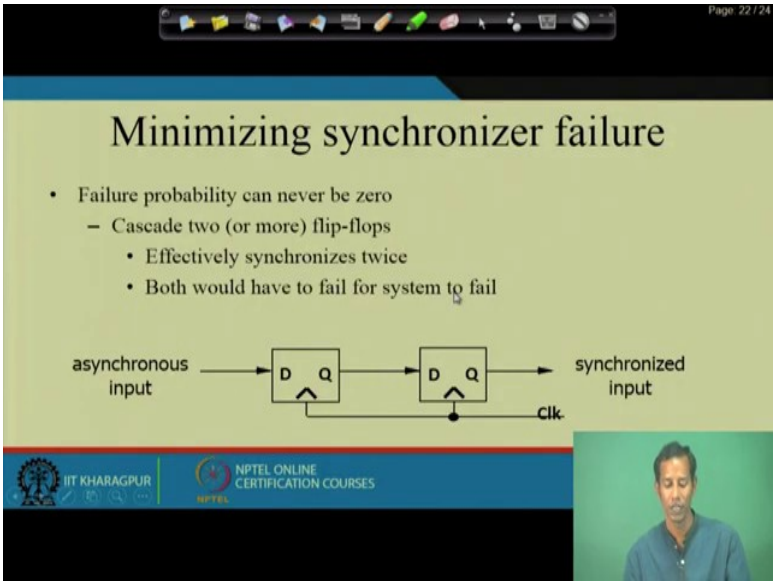
may stand undefined it may also known as this metastable stay or the unambiguous state it can reached to. So, that you cannot recognise the data properly that whether it is in 0 state or whether it is in 1 state which causes this synchronization failure ok.

(Refer Slide Time: 14:12)



So, if we see that what colours this synchronizer may failure is that suppose this is the clock edge and at this particular clock edge if the data changes it is value. So, at that time it does not recognise that whether that the value is 0 or it will pass the values of 1. So, that is named as synchronization failure.

(Refer Slide Time: 14:34)



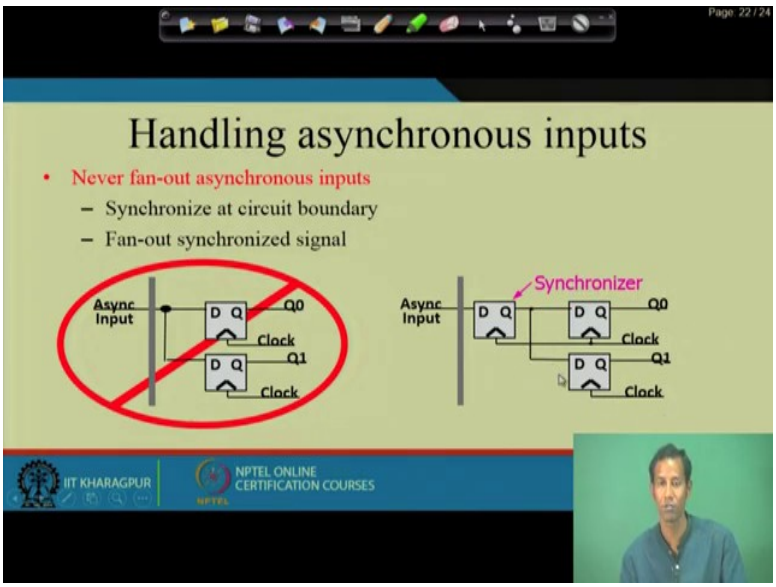
So, how to minimize this synchronization failure is that. So, failure probability can never be 0 so, cascade two flip flops, effectively synchronization use is twice and both would have to fail for system to fail. When both of them are failed so, at that time only the systems will be failed ok.

So, why this? Suppose this input is coming from this asynchronous; that means, asynchronously. So, at that time this first D flip flop will be depending on the clock it will pass to this and then again whatever will be the; that means, because of this asynchronous input whatever changes it occurs here it will first that will be mitigated through this second D flip flop what we used here.

So, while we are doing this this clock domain crossing. So, at that time we use this kind of synchronizer circuit to minimize this synchronization failure, where as we have already said that the real world inputs are mostly asynchronous in nature. So, after actually whenever we are giving the signals to the; that means, our IC the; that means, designed IC. So, at that time it should be passed through a synchronizer so that the inputs all the inputs so, the inputs may be different at different times so, there may be multi bit.

So, all the inputs are basically synchronized and it goes to the system or it goes to the corresponding this; that means, IC at the same time or depending on the same event ok.

(Refer Slide Time: 16:29)



So, how to handle the asynchronous input is there? This never this fan out asynchronous inputs, synchronize at the circuits boundary and then fan out the synchronized signals. So, means what? Suppose, we are having two d flip flops and here we are this asynchronous circuit in; that means, this inputs are coming asynchronous inputs are coming. So, if we do; that means, if we connect; that means, here directly then again it is coming with one another; that means, this path is longer than the this one. So, at that time at the very boundary of itself we need to put this synchronizer ok.

So; that means, this path is also becoming one clock dependent; that means, and then again from here to again it is connected to the next flip flop which is the second flip flop, now depending on that it will be just work as a synchronizer circuit which will be. So, clock domain crossing means; suppose, this is one vault one of the; that means, this is depends on the sample clock and then we are having this design clock or this integrated circuits clock which is the system clock ok.

So, depending on this system clock we are now putting this the extra d flip flop over here to the inputs we are making synchronized with the full system clock, though they are inputs are asynchronous in nature ok.

Thank you for today.