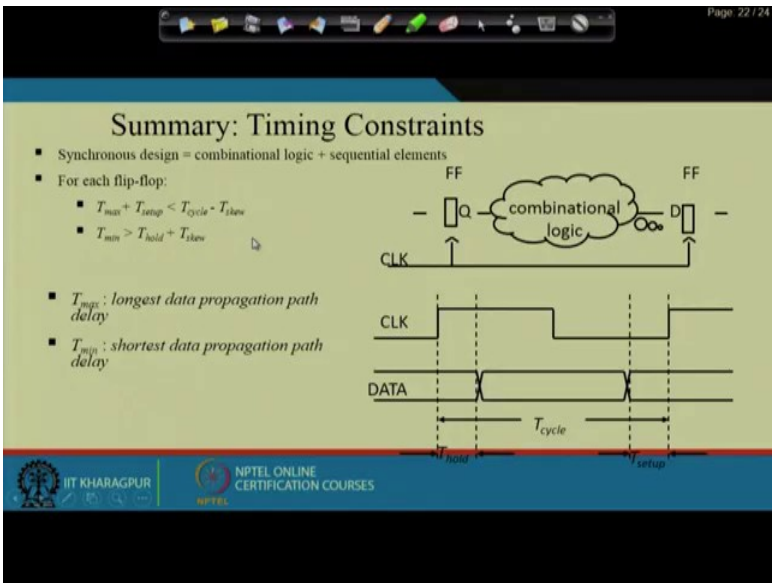
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 52
Timing Analysis Basics (Contd.)

Welcome to the course on architectural design of ICs. So, we are basically discussing and the basics of static Timing Analysis, why it is needed and when it is needed, what are there, whenever, we are considering that is timing analysis. So, at that time what we have to consider so, that our digital IC what we have designed that will not fail when it that means, comes as a; that means, physical chip ok.

(Refer Slide Time: 00:45)



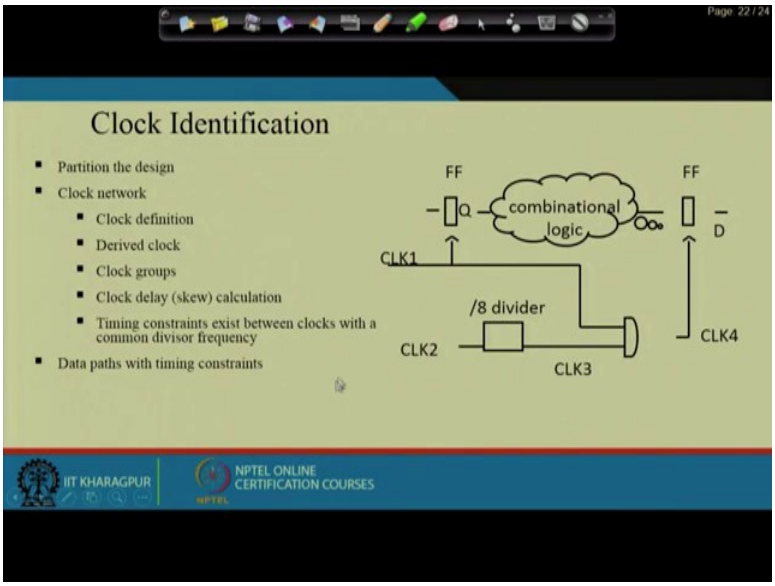
So, we have already seen that to actually calculate the cycle time, we have to consider the maximum propagation delay as well as the set up time as well as the hold time plus if there is skewed clock.

So, the clocks skew also we have to consider. So, as a summary of that; that means, with the continuation of the previous lecture, so, the summary of the timing constraint is that in synchronous design, it depends upon the combinational logic delay plus the logic elements delay. So, for each flip flop the T max which is the combinational delay plus the T setup should be less than T cycle minus T skew. As well as the T min should be greater than the amount of time period of the hold time plus the skew time ok. So, where

T max is the longest data propagation path delay as well as the T min is the shortest data time; that means, shortest data propagation path delay.

So, if we actually look into this particular picture. So, at that time we are having two registers in between you are having combinational logic. So, this is both the flip flops are being basically driven by the same clock. So, in the particular clock so the data at this particular edge it should not be changed. So, after this time requirement; that means, this is the hold requirements after the hold requirement satisfies the data can change its value. So, and this is the setup requirement; that means, this is before the clock is appears this is a time requirement where the data cannot change its value.

(Refer Slide Time: 02:33)



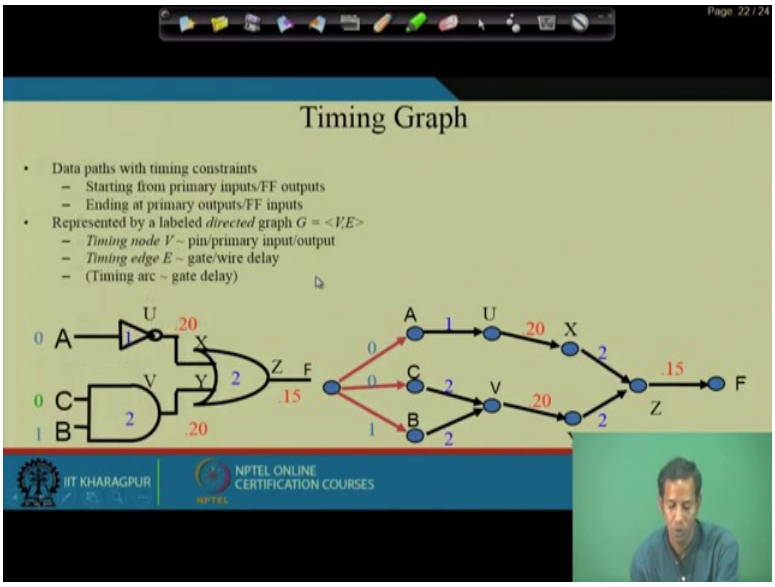
So, now, this thing is that the clock identification. So, the clock identification means, in one particular design, it may happen that the flip flops are basically driven by different clock. So, different clock means, here this particular flip flop is basically driven by clock 1, then this particular flip flop is; that means, driven by one another clock to which is divided by 8 and then we are having this clock 1 and clock 2 ended which is basically going to this particular flip flop. So, this is some kind of gated clock, we have used in the case of flip flop 2.

So, in this in the clock network analysis in any digital IC that is much more important. So, whenever we do this clock network analysis. So, at that time we have to defined the clock very fast, then we can generate the derived clock; that means, if one clock is

basically derived from another clock. Then we can create the clock groups as well as we have to; that means, calculate the amount of clock network delay or there which is nothing, but the skew and then we have this timing constraint exist between the clocks with a common divisor frequency which is this clock divider operation.

Then actually this data path should be constraint depending on the timing constraint what we have calculated on these different 4 different clocks.

(Refer Slide Time: 04:21)



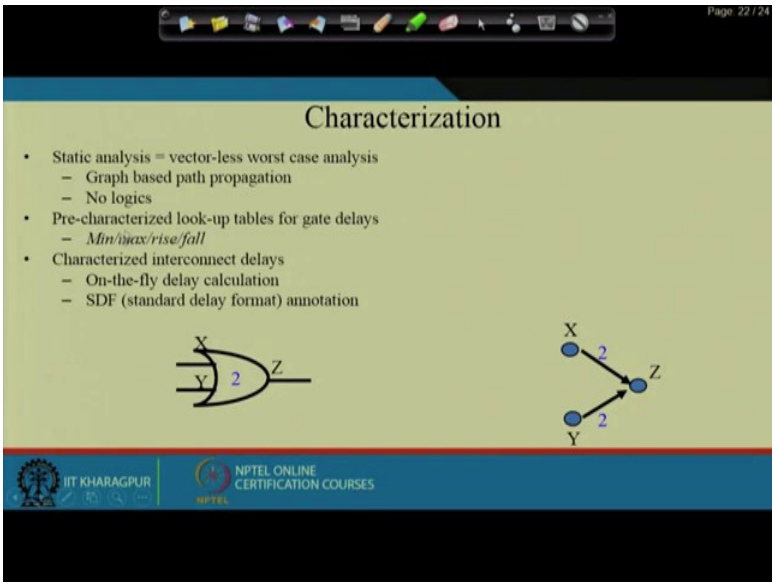
So, how we can calculate or how we can actually any EDA tool basically analysis the timing constraint on a particular circuit. Suppose, this is one circuit which is being implemented in digital IC, so, what is there it is having A B C, 3 input variables which are basically consists of the here this 0 0 1, this corresponds to the input delay which; that means, comes at A B and C. And this 1 2 2 these are the delay amount which corresponds to each of this gate delay. And then we are having this U to X V to Y which is 20 20 and here Z to F they are basically also level to the corresponding this particular paths timing information.

So, then if you see that data path with timing constraint starting from primary inputs or flip flops output, ending at primary outputs or as a flip flop inputs. Represented by a level directed graph which is G consisting of this V and E. Where this timing node V is the pin or the primary inputs or the outputs and E is the gate or the wire delays. So, then

from this particular circuit now, if we make the corresponding timing arc so it may be seen like this well this is the starting where from; that means, this is the source node.

So, here from this to A so here this is let us say this is one dummy node. So, from to A this is 0 this is 0 this is 1, then again A to U, A to U which is little considering this one C to V C to V is the 2, then B to V again this is 2 because this is the input to C to V B to V. Then U to X U to X corresponds to this 20, V to Y corresponds to 20 and then again this X to Z Y to Z corresponds to 2 and 2 and this Z to F corresponds to 15. So, from this particular circuits information, the tool basically corresponds this directed graph based on the timing information what we have achieved here ok.

(Refer Slide Time: 07:24)



So, then how we can characterize this, this characterized one particular circuit or one particular gate. So, the static analysis is the vector less worst case analysis which basically consists of this graph based path propagation, where no logic information is stored. This pre characterized look up tables for gate delays pre characterizing look up tables for gate delays means, because this gate delay information that is fixed suppose for this X or gate sorry, for this OR gate this gate delay is two amount of; that means, 2 time 2 amount of time period. So, then again so, from X to Z it will take 2 Y to Z that will 2.

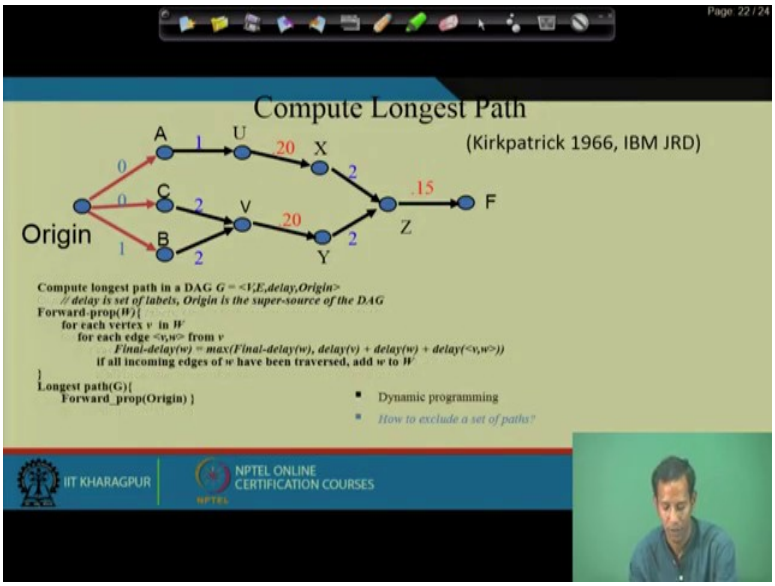
So, the where from you will get this information that this comes from the particular library which we use in the digital circuit design. So, if we use let us say 90 nanometer technologies. So, at that time this particular gates delay that will be let us say 1

nanosecond, if we use 180 nano; that means, nanometer technology. So, at that time the gate delay should be more which will be 2 nanosecond of time. So, that information is technology specific. So, depending on their, those where this will be stored this will be stored in the lookup tables for all these gates delays information and for the interconnect delay it basically calculates on the fly ok.

So, on the fly means depending on the path length so; that means, what is the wire length intermediate wire length depending on that its capacitance and resistance value this delay calculation has been calculated and whenever, we are in the previous slide whenever we connected that from the interconnect path, if you do let us say its corresponds to 20 amount of time period or its corresponds to 2 time period. So, it depends upon that what is the wire length and which metal layer we are basically using to connect them.

So, depending on all this information it calculates the delay for those particular paths on the fly and we have to for that to calculate this interconnect delays, we have to consider the standard delay format annotation. So, this is this SDF annotation is nothing, but the corresponding RC extraction of your particular circuit. So, that you can get interconnects delay very specifically depending on the library what technology library you are using ok.

(Refer Slide Time: 10:22)

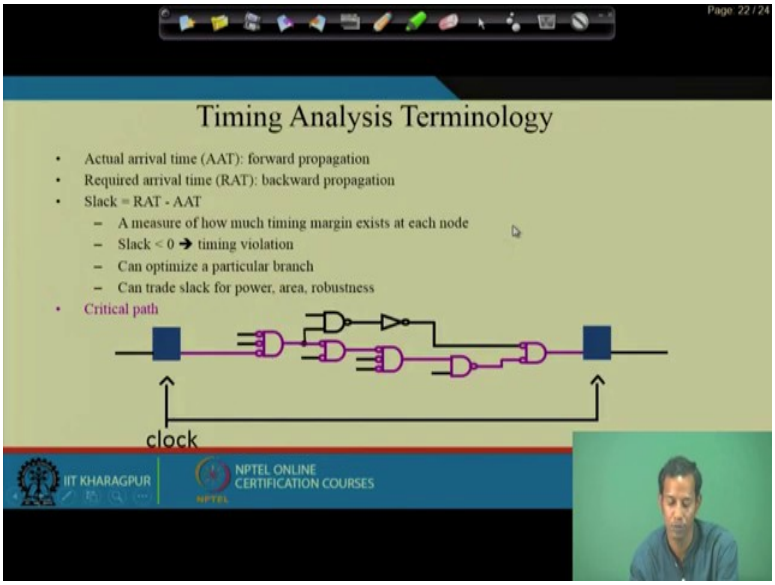


So, then depending on this now, you can make the corresponding algorithm or you can make the corresponding that data flow data flow graph and from there you can calculate.

So, you got you have developed this graph from the particular circuit. Now, you solve this; that means, this you write this algorithm. So, that it can calculate that for each of these paths which one is the maximum longest path it takes, but from reaching from this particular source to the end which is a point F and it gives you the what is the long; that means, critical path of your circuit ok.

So, now, here actually there as there are like 1 path 2 path and 3 paths. So, how to exclude a set of paths? So, there may be walls; that means, except the critical path the other paths are also there. So, there will be this maybe there will be multi cycle path or there may be one false path. So, what is this multi cycle path? What is this false path later on we will again see what are the meaning of those?

(Refer Slide Time: 11:50)



And when we will use or when we will define that this is the false path this is a multi cycle path. So, you do not you include or you may exclude this path in the timing annotation so that we will also again see.

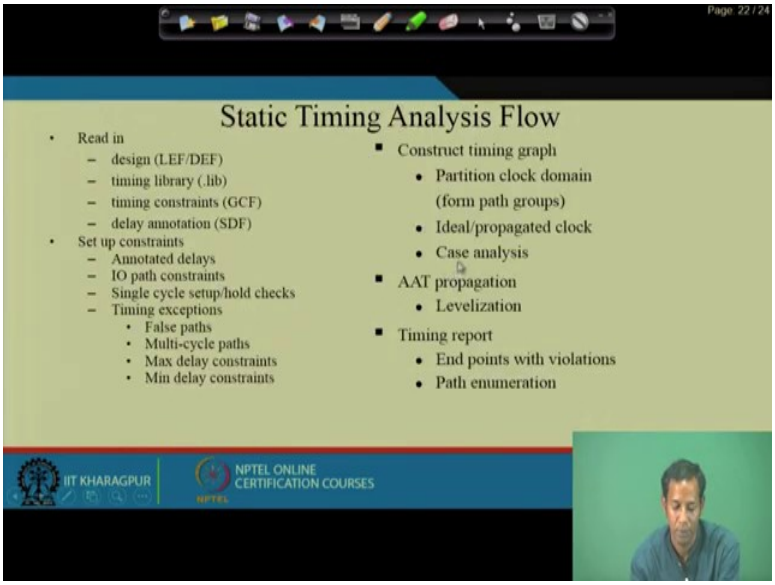
So, in timing analysis terminology, we are having this terms which is this A A T actual arrival time which is nothing, but with forward propagation this required arrival time which is this R A T backward propagation time. So, the slack is the difference between this required arrival time minus actual arrival time. So, what does it means is that slack all the time it should be near to 0 or it should be in positive value. Why? Because here you see a required arrival time means, that is the actual requirement time requirement of

your circuit ok. And this actual arrival time, if it is more required let us say required is let us say 2 and actual arrival time if it is 2.1.

So; that means, in actual of this particular hardware when it runs. So, at that time it is failed to do this required arrival time means, it is just calculated value. So, on that particular case, it will not get that it is not satisfying the condition that it should be arrived within 2 nano second of time as it is in actually it is taking more time which is 2.1 nanosecond of time.

So, that is why at that time in any digital circuit design slack should be 0 means at that time both are same. So, required time and the actual time both are same that is perfectly fine designed, but positive also its ok, but negative means there is problem so; that means, you are required time will be changed to the actual arrival time. So, that is actually we calculate, whenever we analyze this we have to calculate this whether the slack is positive or negative.

(Refer Slide Time: 14:16)



So, depending on that we can say that my design will work perfectly or not.

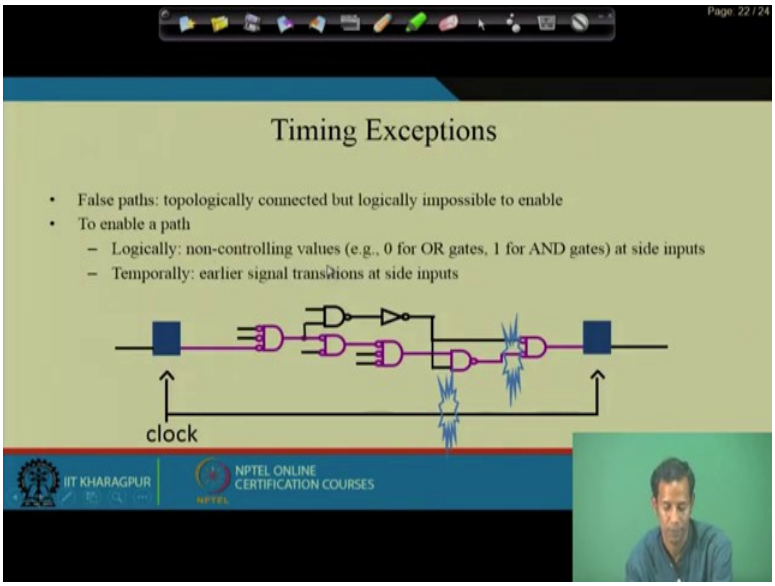
So, whenever we do the static timing analysis. So, this design can be read in in the format of this LEF or this DEF this it requires this timing library information, it requires the timing constraint which is the extension will be GCF the delay in annotation which will be either SDF. Then the constraint which we can set up is that this annotated delays

the IO path constraint the single cycle setup or hold checks or the timing exceptions like this false path, multi cycle path, maximum delay constraint and the minimum delay constraint.

While we are constructing the timing graph so, at that time we can partition for different suppose, we have having multiple clocks in the design. So, at that time we can partition the clock domain we can create the ideal or the propagated clock or the derived clock what we have said earlier and we have to do the case analysis.

So, A A T propagation in the AAT propagation, we have to do this levelization. A A T propagation means, this actual arrival time propagation we have to do this levelization and in timing report, we will see all the end points with the violations if there is any and with all these paths enumeration; that means, all the information about each of the paths which are already there in your circuit design.

(Refer Slide Time: 15:56)



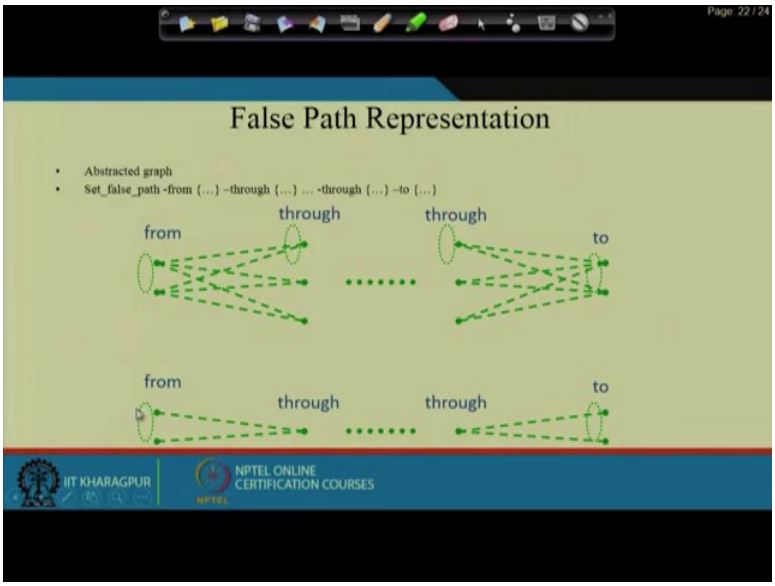
So, now, this timing exceptions so timing exceptions what it says is that false path, multi cycle path. So, what is this false path? False path is the topologically connected, but logically impossible to enable means what? Suppose, here actually if we consider this we are having two path one path is this another path is this one.

So, to enable a path logically this non controlling values which is that is 0 for or gates and 1 for and gates at the side of this inputs and in temporarily, if we want to enable the

path in the earlier signal transition at side inputs. So, means what actually in false path in and gate if one of the input is 1. So, at that time the other; that means, the output will be just the same; that means, if it is 1 then if it is 0 it will be 0 if; that means, if it is A and it is 1. So, it will be just A in case of or gate if this is 0. So, it will be just A ok.

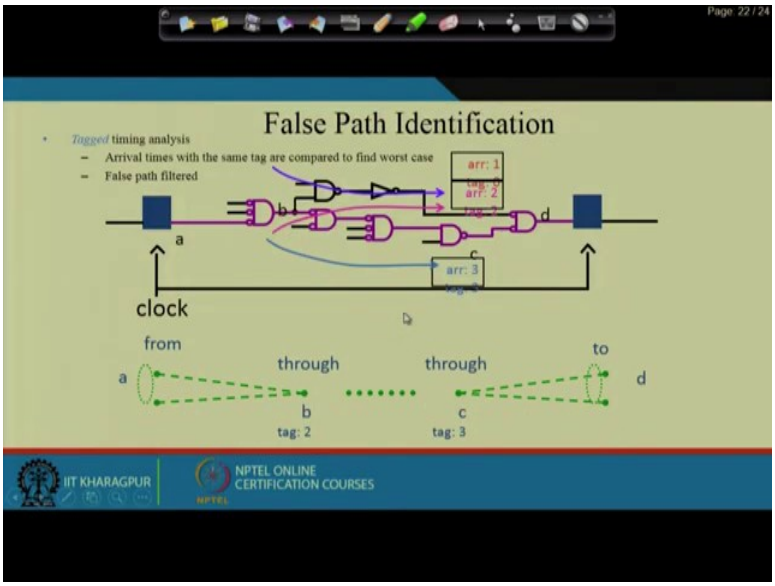
So, topologically, it may be connected, but logically it is it may seen that it is not connected to the input to that; that means, to this particular longest critical path.

(Refer Slide Time: 17:33)



So, if we see that on how we can; that means, to represent the false path is that. So, suppose we have this is this is from and this is through this is also through this is to. So, not this one.

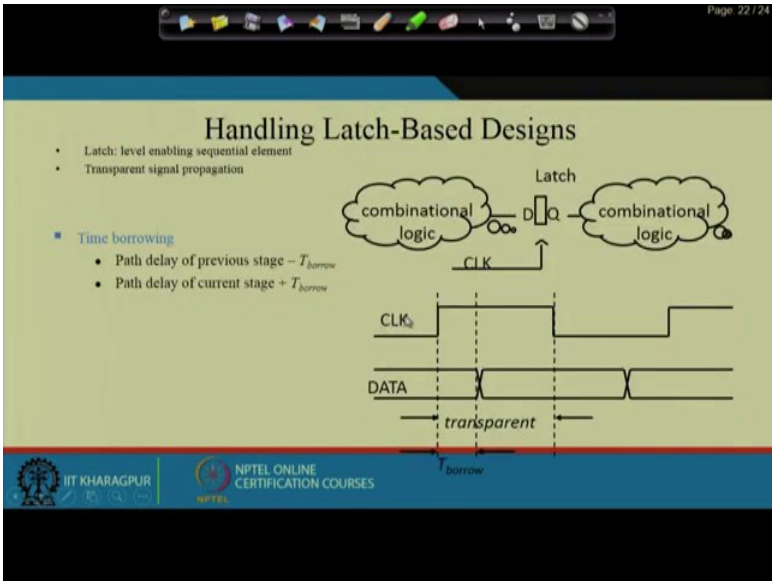
(Refer Slide Time: 17:52)



So, how we can this identify the false path is that suppose here we are having this particular one path from here one from there and another path with this.

So, in arrival 1 this is 1 with tag 0 and that in the in the second one this one we are having this arrival of 2 with the tag of 2 in this we are having this arrival of 3 with tag 3. So, now, we have basically formed the graph from this particular circuit and then, we calculate that the tag timing analysis basically relates to arrival times with the same tag are compared to find the worst case and the false path are filtered ok.

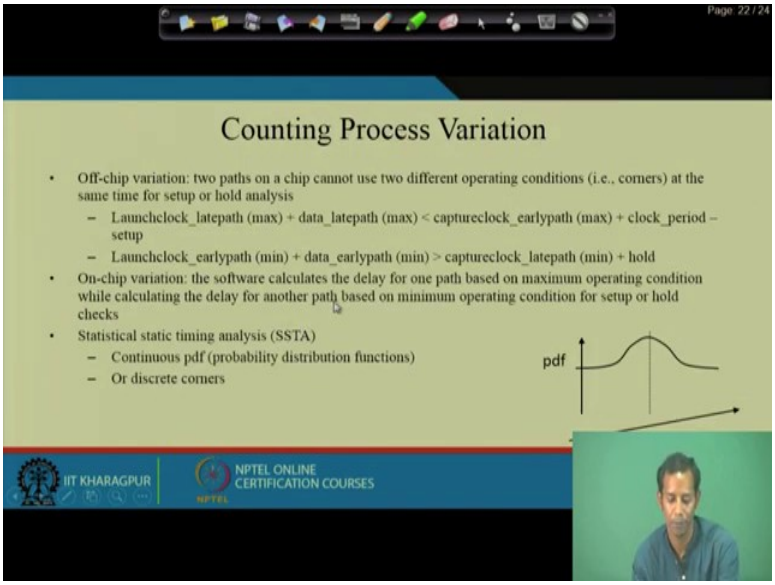
(Refer Slide Time: 18:57)



So, in flip flop, the flip flops are basically edge sensitive ok, but the latches are level sensitive.

So, how to handle the latch based design then? So, the latch based design is basically level enabling sequential element and it is basically transparent in signal propagation. So, in case of latch, we have this time borrowing. Time borrowing means what? Whenever this clock is 1 at that time the latch is transparent; that means, it can it can pass the value what is the; that means, the data value it takes as the input. Whenever there is a 0 at that time it should not pass the value which contains the in the form of data ok. So, that is why you see whenever this clock is 1. So, at that time this is basically the latch is transparent whenever this is 0 the latch is closed.

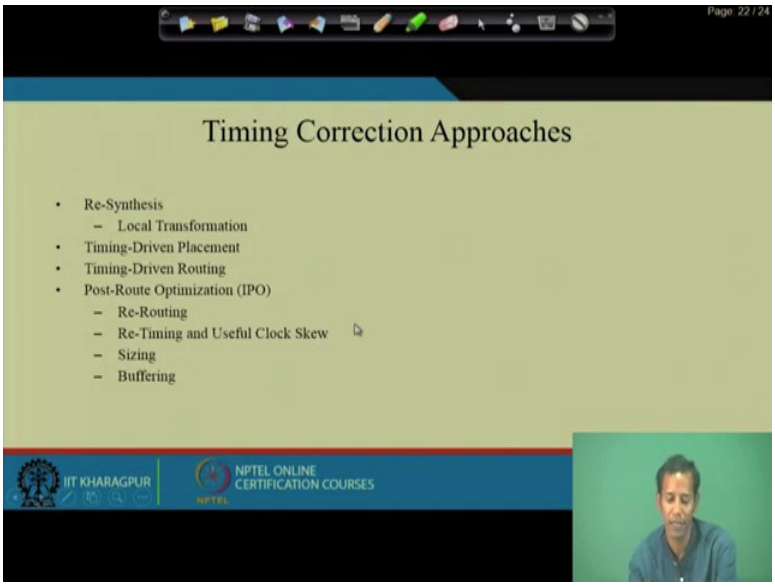
(Refer Slide Time: 20:14)



So, actually the process variation has an important role in this timing analysis. How? This off chip variation, so, what is this off chip variation? When two paths on a chip cannot use two different operating conditions at the same times for setup and hold analysis. So, at that time what will happen? That means, suppose we are having two different chips they cannot be having two operating condition; that means, one is running at let us say 0 degree centigrade and another one is let us say running at 85 degree centigrade that may not be used while we will calculating this set up time and hold time analysis.

Because at that time this information, we are retrieving from the library that library information will be different. So, the calculation will be totally wrong. So, this is the; that means, calculation; that means, equation for different of this timing analysis ok. So, this depends on this off chip variation as well as this on chip variation, along with this statistical timing analysis which contains this continuous PDF this probability distribution function or the discrete corners.

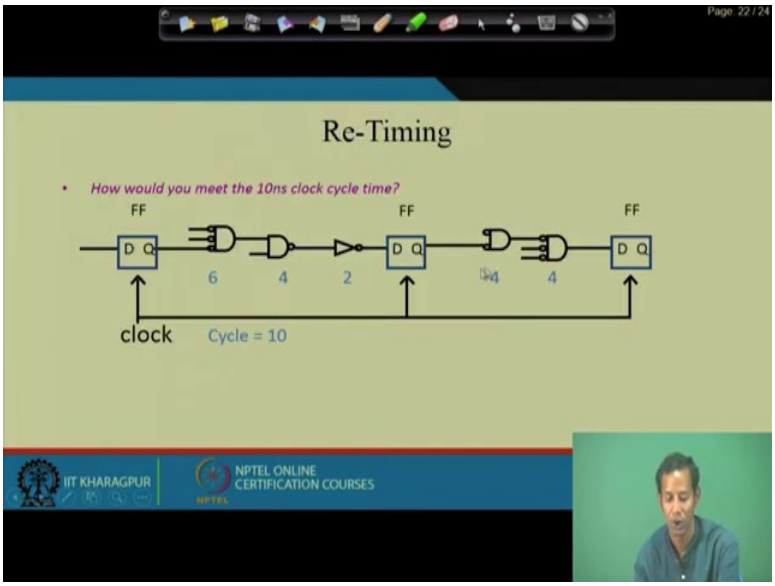
(Refer Slide Time: 21:45)



So, the corresponding timing correction approaches are based on the re synthesis this timing driven placement or this timing driven routing.

In the post route optimization ok, we are having this rerouting, retiming and useful clock skew sizing and the buffering ok.

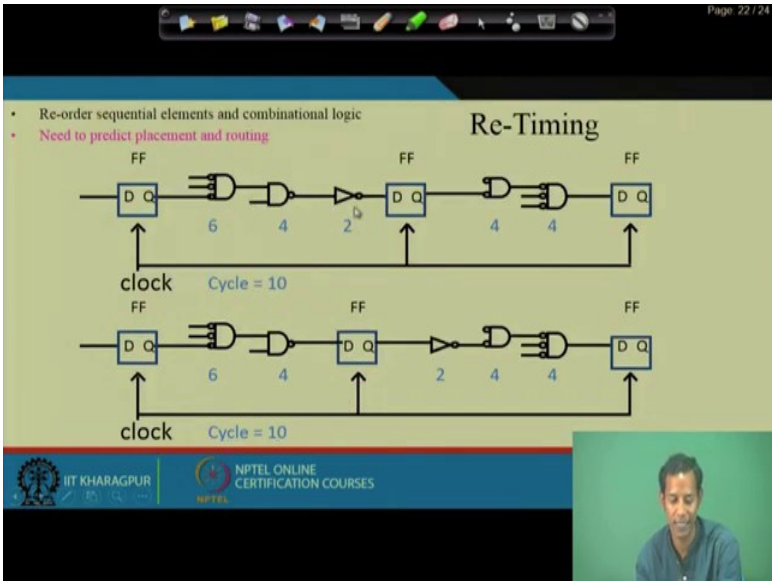
(Refer Slide Time: 22:09)



So, what is this? So, retiming what is the meaning of retiming is this. Suppose, we are having one circuit like this ok. So, here we are having this is the delay of this gate is 6 delay of this is 4 and the delay of this is 2. So, 6 plus 4 plus 2 the total; that means, path delay for this is 12 and here we are having two gates one is this 4 another one is 4 so 8. Now, how you can meet the 10 nanosecond clock cycle time requirement ok.

So, the clock cycle requirement for this is 10. So, if I set this clock cycle requirement as 10. So, at that time this particular requirement is 12. So, it will not work.

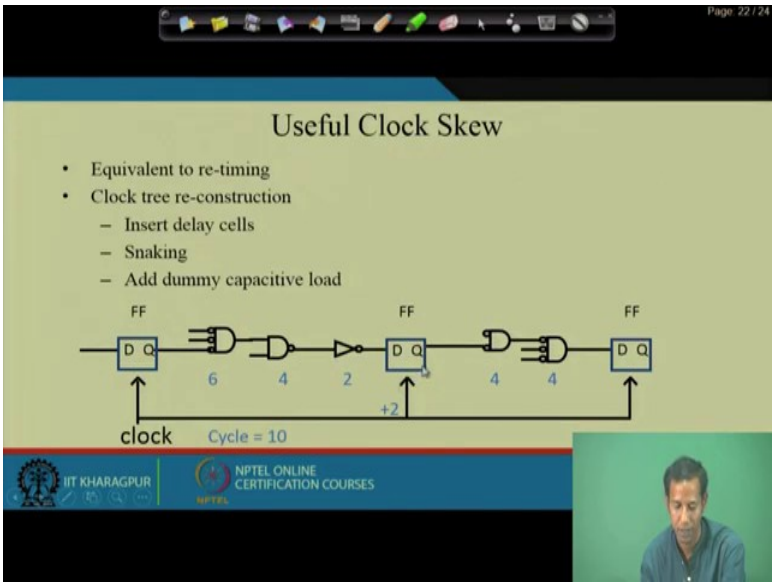
(Refer Slide Time: 23:05)



So, how you can make it work? So, what you can do you can just shift this inverter to the other side of the flip flop. So, at that time this path will become 6 plus 4 10 and 2 plus 4 plus 4 is 10. So, actually do you see any problem here? The problem is that, actually we are doing what? We are doing retiming. Retiming means, we have just shift the registers in the just; that means, we have placed the inverter just after the register. So, that every; that means, it becomes balanced the registers paths because the critical path between the register becomes balanced.

So, so, in this case actually as we have already done this; that means this change from to this. So, now, initially the requirement of this particular path was 12 now it becomes 10. So, this is also becoming 10 so; that means, now the clock cycle requirement of 10 that is satisfied.

(Refer Slide Time: 24:30)

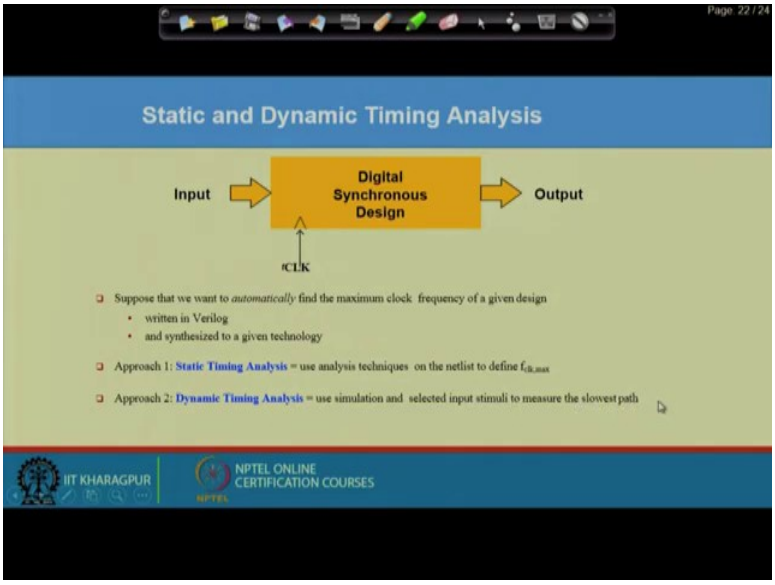


So, now this useful clock skews. So, what is this useful clock skew? That means, this useful clock skew is equivalent to retiming, this clock tree reconstruction it insert delay cells and add dummy capacitive loads.

So, whenever this particular D flip flop they are basically added with two amount of this clock skew. So, at that time it will also meet the clock cycle time requirement of ten ok. How? Basically; that means, as you can you can; that means, remember the corresponding equation that this will be T max T max plus T setup that should be less than equals to T cycle minus this the skew. So, if the T cycle at that time it will be 10 as

it is 12. So, at that time this use this because of this plus amount of two skew we can make this cycle time to validate or to follow that though the path corresponding path is related to twelve.

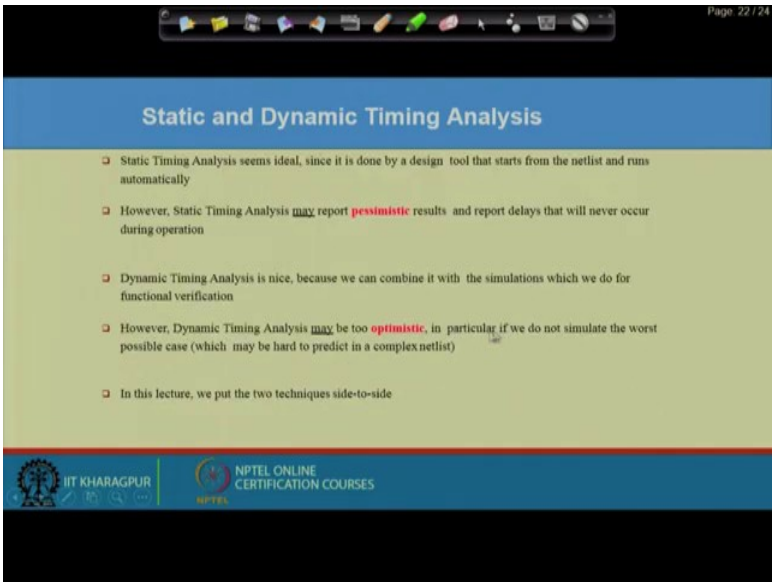
(Refer Slide Time: 26:01)



So, now what is this actually this digital in the synchronous digital circuit design ok, we are having this static and dynamic timing analysis. So, what is that? Is that suppose, we are having the bunch of inputs and we are having the bunch of outputs, we are having one particular single clocks which is gone going to every of the sequential elements which are present in the digital synchronous design. So, this suppose that we want to automatically find the maximum clock frequency of a given design which is written in verilog and synthesized to a given technology.

So, the approach 1 is the static timing analysis which uses the analysis technique on the netlist to define the f clock max and the approach 2 is the dynamic timing analysis which uses the simulation and selected input stimuli to measure the slowest path.

(Refer Slide Time: 27:00)



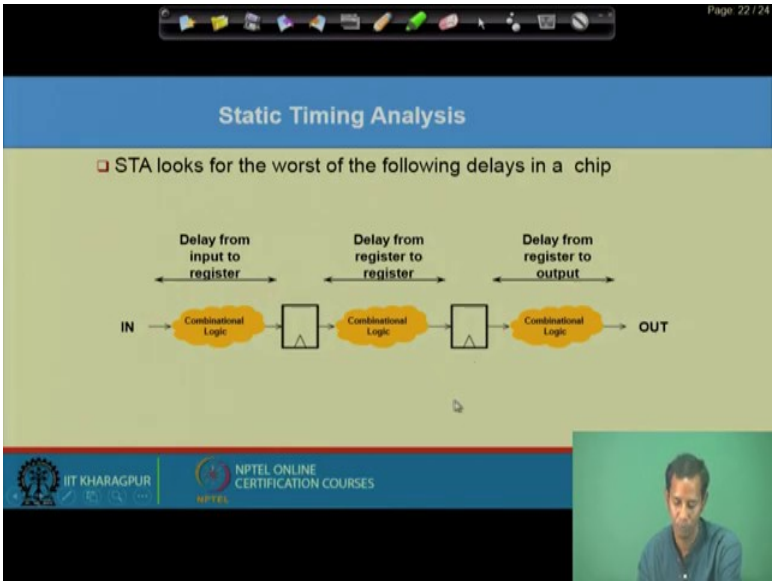
So, static timing analysis basically it seems the ideal one and it since it is done by a design tool that starts from the netlist and runs automatically; however, the static timing analysis may report pessimistic results and report delays that will never occur during operation.

On the other hand dynamic timing analysis is nice because, we can combine it with the simulation which we do for functional verification and; however, the dynamic timing analysis may be too optimistic in particular, if we do not simulate the worst possible case and in this lecture we put the two techniques side to side ok. Means what? What is the difference between these static timing analysis and dynamic timing analysis is. In static timing analysis, we do not consider that depending on the inputs whether that is 0 or 1 how the circuits basically change its behavior? That means, suppose for if we calculate, if we have; that means, make the dynamic; that means, if the timing graph. So, at that time depending on the inputs 0 or 1 or the interconnect paths whether that is the value is 0 or 1.

So, at that time whenever, we calculate the corresponding gate delays are same, but the net delays whether it is passing the 0 or whether it is passing the 1. So, at that time what amount of delay it is considering in-actual. So, there if I consider that so at that time it is dynamic timing analysis if I do not consider just depending on the library information, if

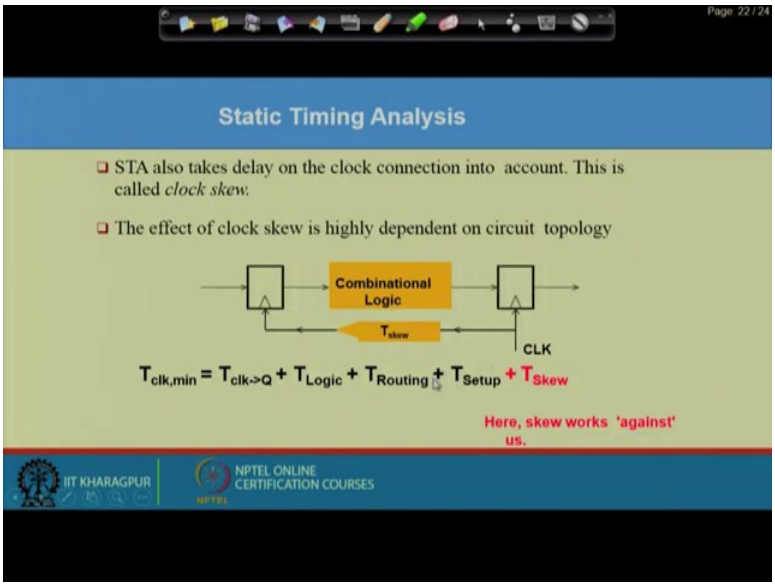
we calculate depending on R and C value, what is the delay for this interconnects? So, that is the static timing analysis.

(Refer Slide Time: 28:56)



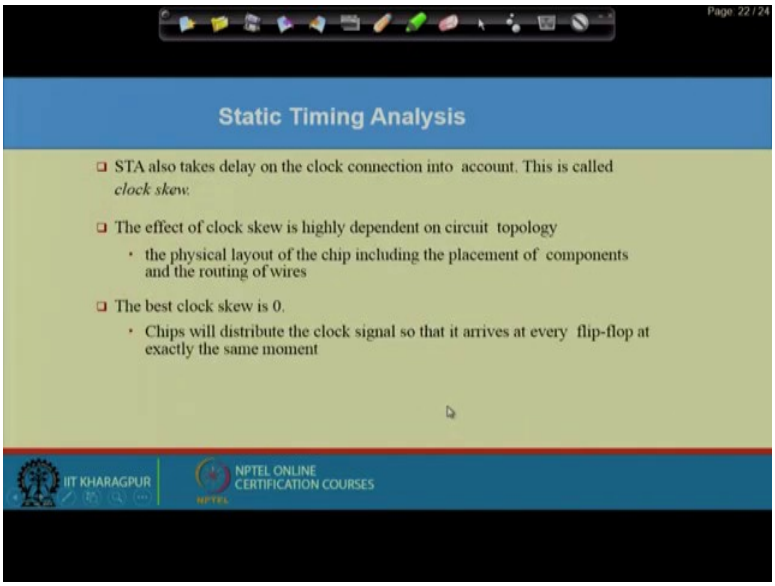
So, this static timing analysis again actually, it is the same what we have already seen.

(Refer Slide Time: 29:08)



So, this is the skew actually sometimes, we use the skew for benefit in the; that means, circuit.

(Refer Slide Time: 29:18)



And this static timing analysis takes the delay on the clock connection into account this is called the clock skew and the effect of clock skew is highly dependent on circuit topology. The physical layout of the chip including the placement of the component and the routing of the wires and best clock skew is 0 best clock skew is 0 means, actually there is no such difference between the clock is appears to the different or the various sequential elements which are present in your digital circuit design.

So, and why this effect of clock skew is basically it highly dependent on the circuit topology means, where we have placed from the clock pin to the corresponding all the circuits; that means, sequential elements where it is being placed depending on that; that means, how much path the clock has to traverse depending on that the delay is or the amount of skew we can calculate.

So, if the clock has to; that means, traverse a long path. So, at that time the skew requirement is more, if the if the clock is directly connected to a sequential element; that means, the path is smaller so at that time skew amount is lesser. So, that is why this effect of clock skew is highly dependent on circuit topology. So, for today this is it again next day we will discuss some of the other things on static timing analysis.