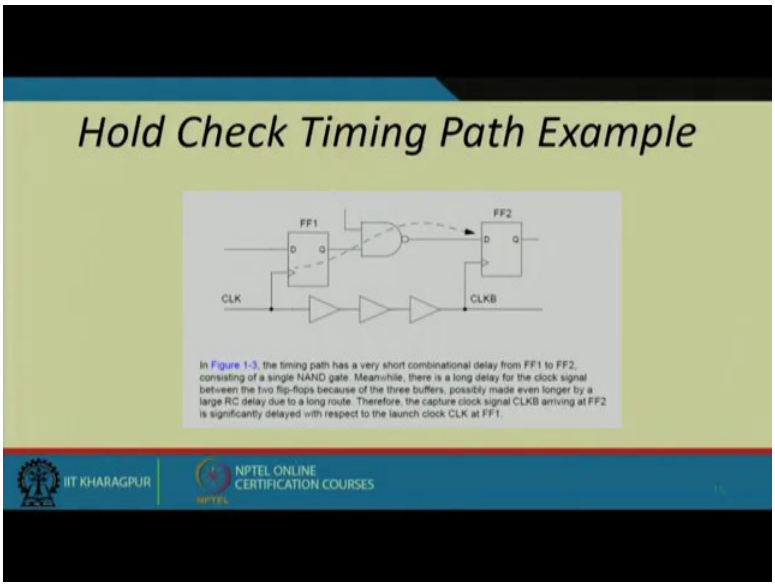
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 55
Timing Issues in Digital IC Design (Contd.)

Hello everyone, welcome to the course on Architectural Design of ICS. So, we are basically discussing that the Timing Issues which are there in while we are performing the digital IC design. So, we have seen that there are some terminology which are associated with this a clock source or the clock pin. And whenever we are considering the timing analysis means we are basically more careful about the clock source or the clock which we have used in our circuit.

So, there must be some like this clock latency, then the input delay output delay and then clock jitter, clock skew. So, these are the terminology skew which are which we have to consider when we are doing this static timing analysis. And, we have also seen that why this static timing analysis is much more preferable than the dynamic timing analysis.

(Refer Slide Time: 01:13)



So, again we will see that how; that means, in the previous lecture we have seen the set up how we perform the set up timing check in the corresponding circuit. So, now, we will see the hold time check for any particular circuit.

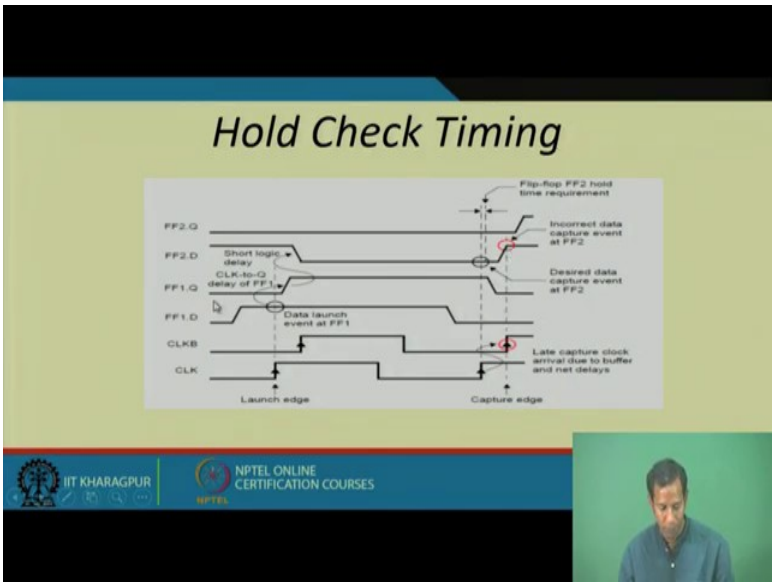
So, hold time circuits is that if we take one of the example the same circuit. So, they are actually the clock pins where like clock is connected directly here and then to the second flip flop. But in actual case in actual it does not happen or it does not look like that. Why? Because whenever you will synthesize you will be having or you will be actually if you are having the digital IC core. So, at that time you will be having this entire particular sequential element, it is not that it is placed on a particular place.

So, it will be distributed all over; that means, all the corners of the particular core. So, at that time and your clock source that will be core through one particular location. So, now, from that location now you have to distribute the same clock to every of the sequential elements which are present in your circuit or in your core. So, at that time the distribution of this basically requires the clock tree network or clock tree synthesis.

So, in clock tree synthesis whenever we actually the as this particular example if this particular this is the location of the clock pin, the path which is traversing to reach this particular flip flop and the path which is needed to reach at this clock pin of this particular registers they are not same. So; that means, the delay for this particular clock and this particular clock there will be one difference. To increase the corresponding strength of the clock, we put some buffers on the clock tree path. So, that every sequential elements which are present on the corresponding digital IC gets the clock signals of same strength ok.

So, that is why we put some of the buffers in the clock tree path and it reaches the farthest sequential elements ok. So, at that time so, because of that we will get the clock signals with very high drive strength or it that is the corresponding clock signals will be much more clear. It will not be with; that means, as there it has to traverse a long path it is not that the clock signals will be much more degraded. So, it will be much more on the pure side, if we put the buffers, but whenever we put the buffers on this clock tree path. So, at that time it causes the hold time violation.

(Refer Slide Time: 04:27)



So, why hold time violation is that, now if we consider how this hold check timing; that means, occurs at the static timing analysis, that we will see is that suppose this is the clock source ok. Now, here again when these particular things this particular edge occurs. So, at that time we get this is the clock and this is the clock B. So, clock B is basically delayed. Why it is delayed? Because as we have already put some of the buffers over here. So, depending on the delay of those buffers this particular clock so, signals will be delayed and which will be act as a clock B.

Now, this clock A if this clock is connected with flip flop 1. So, the input the input of this flip flop 1 is this. So, at this particular edge it finds that this is the capture edge. So, at this particular edge it finds that the D value input value is D. So, it will be just after some time delay, it will be Q will be gone to high then again what will happen with the second one the second flip flop.

So, at that time as this is already sifted so, at that time at this particular edge this is the capture edge for the second flip flop. So, at this particular edge there will be some incorrect data which will be captured at flip flop 2. Why? Because it depends upon this flip flop 1 and then this is the out this is the; that means, this Q has to pass through this NAND gate and then again it will reach to the flip flop D. So, if you consider this flip flop D then it will be just having this corresponding output of the; that means, NAND

gate. And here there are some of this there is very short difference or because of this delay of this particular NAND gate. So, after that again it is working like this.

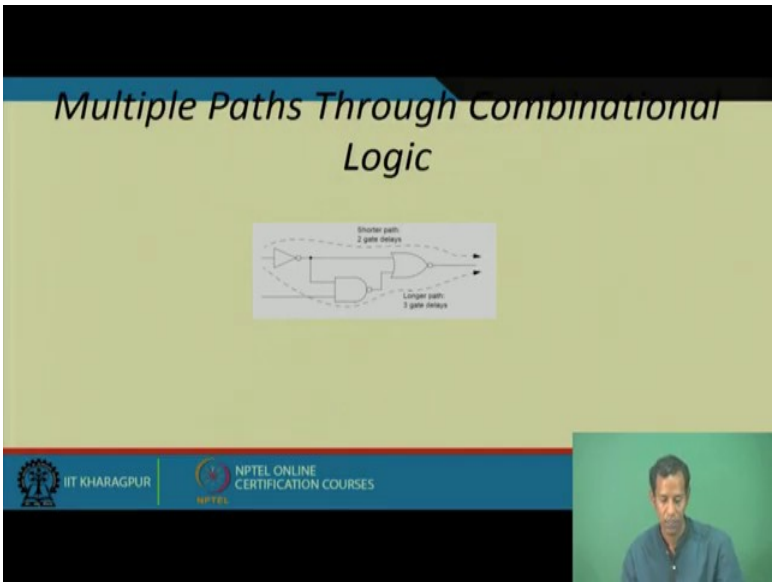
So, whenever we are considering this particular edge, at that edge as this data is already changing; that means, what we know that whenever we will consider at that time whenever we find the edge. So, at that time before the clock edge arrives and the after the clock edge arrives the data should not be changed. Again, as this particular edge the data is changing. So, it cannot recognize the data properly. So, at that time in flip flop 2 you will be having a hold time violation ok.

So, that is why if this kind of things happens at this particular edge this data is changing its value that is why you are getting the hold time violations. Otherwise if this data has changed before sometimes or after sometimes at that time you will not get any of this hold time violations ok. So, at that time it will be ok, but it may happen that some of the paths it can or it may get the situation something like this, where you will get the hold time violation because of putting this extra buffers which creates this clock B as the delayed version.

If you see at this particular edge, if I consider the flip flop tools input. So, at that time there is no such; that means, problems it is not changing the data at this particular edge. As we have put the buffers so, that is why it comes late and there the corresponding in; that means, the output from the combinational logic which is the input to the registers of flip flop 2 that is changing so, that is why it is creating the problem.

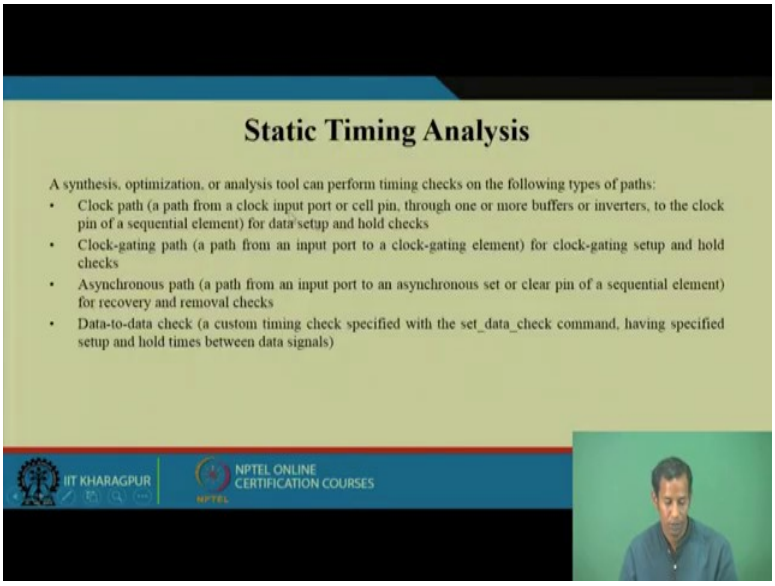
And you are getting the violations ok. So, at that time, how you can fix this violation either you can put some extra buffers to delay the clock edge or if you have put more of the buffer. So, at that time you remove some of the buffer. So, that this clock edge occurs sometimes before ok. So, that is depending on your situation or your requirement whenever you are working with this ok.

(Refer Slide Time: 09:20)



So, then again if you consider one, but suppose you consider one combinational circuit like this then we are having; that means, two different paths from input to the output ok. So, here actually if we, if you consider that there is one path which is coming or which is going from this to this. So, here you are having shorter path, which is only two gate delays, but in this particular path you are having longer path and you are having three gate delays.

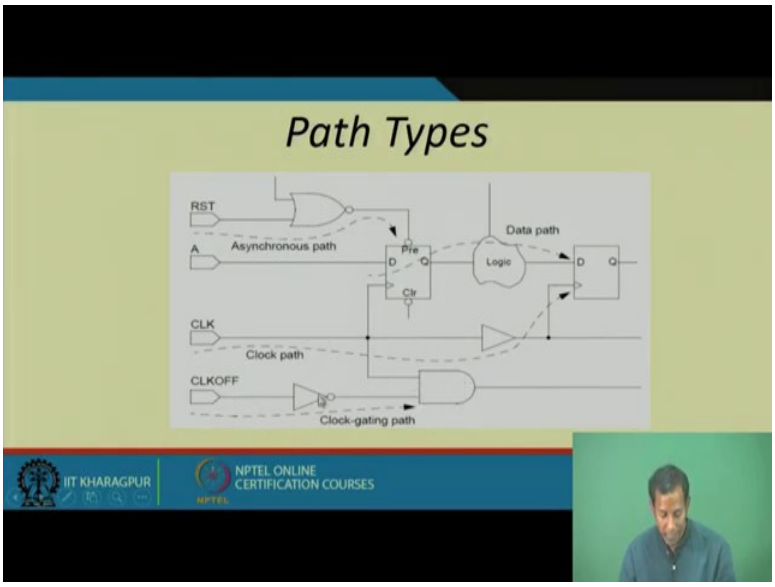
(Refer Slide Time: 09:57)



So, why to consider is that, there may be this kind of paths more than one path will be available from input to the output. And, every of this timing information of every of this path you have to calculate using static timing analysis. Unless and until if you miss any one of this particular path and it creates or it associate with the corresponding violation. So, at that time your chip will not work it may fail to perform successfully ok.

So, at that time all the money all them; that means, effort you what you have put that will be all in vain. So, that is why to double check or to; that means, make sure that the wall; that means, the IC what we have designed that is working perfectly fine. So, there might be several paths like this clock path, then clock gating path then asynchronous path and then data to data check path. So, we will consider each of these paths and what is; that means, how what analysis we have to do on this path that we will see.

(Refer Slide Time: 11:18)



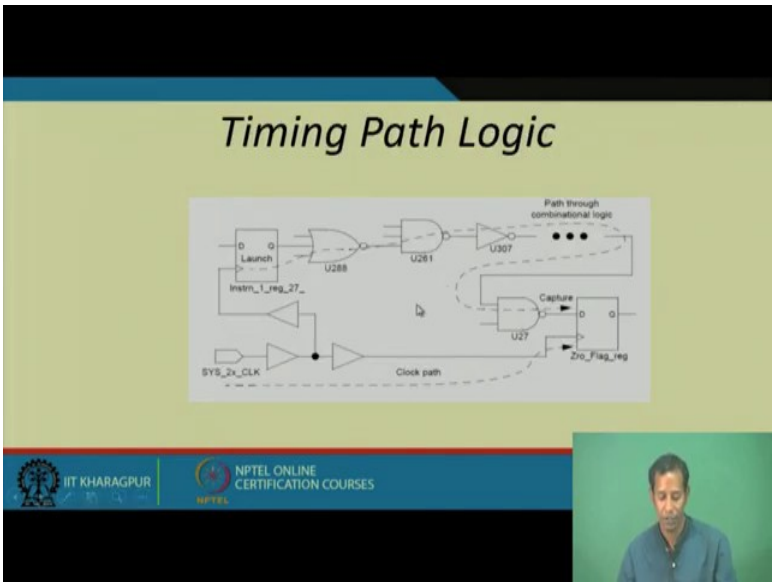
So, the paths if we if I say that ok. So, if I say that there are there may be asynchronous paths which are related to clock, there may be clock gating path there may be only single simple clock path there may be data to data checkpoint data to data check path. So, what are these? If I consider the logic something like this so, here you see that reset signals is basically anded sorry or nor with some other logic and it goes to preset. So, this is nothing, but the asynchronous path which affects the corresponding flip flops operation. And then again this clock is basically connected here and then by passing one buffers to the second flip flop.

So, this is nothing, but the clock path. Now, again if you are having some; that means, your clock is basically Anded with some other logic ok, with this is known as clock gating path. So, mainly this for low power technique low power technique means what? Suppose one if I actually if I design this particular circuit and if I what I said is that clock is basically free running; that means, it is continuously running. So, if I just run this clock in some case the design may be or may not be actually consider one case is that suppose the A value is 0 it is not changing its value it is continuous; that means, A is fixed let us say 0 or 1.

Now, the clock is continuously running. So, though this particular data is not changing its value, it is basically in the idle state. So, because of this clock running the whole circuit will run continuously. So, if I run the circuit continuously, it will consume the power. So, as this will consume the power. So, to reduce this power if we use some of the; that means, technique that if I find that at this particular a condition if I find any idle state so, at that time the clock will be stopped ok. If it is in the running mode at that time or in the wake mode at that time only this clock will be activated and it will be run continuously ok.

So, for that type of logic if we want to implement, we need this clock gating path or clock gating circuit we need to make it for useful for low power application. And then what is the data path? Data path is the; that means, the input; that means, from the flip flop 1 data input to the flip flop 2 data input. This is nothing, but the data path means what there is no such deal; that means, all the delays which are associated here they are basically purely combinatorial.

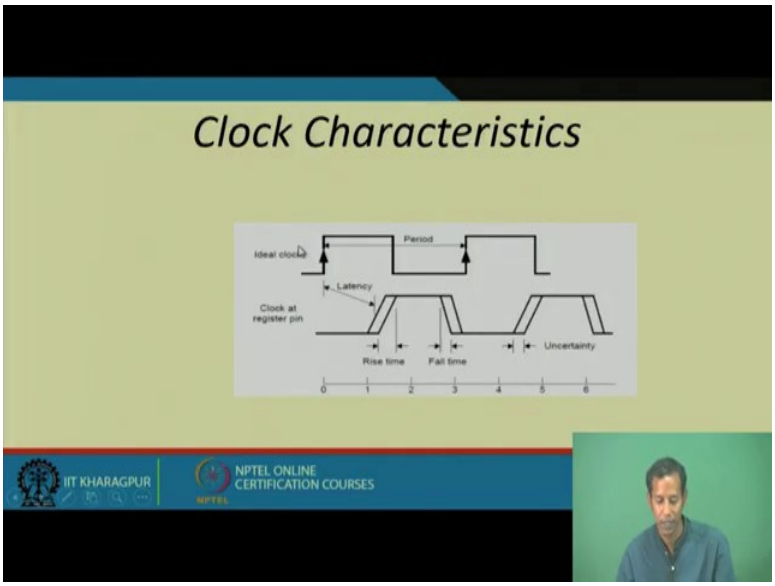
(Refer Slide Time: 14:37)



So, then again if I consider one and; that means, another example is that suppose there I am having the system clock. Now, this is going through one of this D flip flop then again with some logic and again it is going to some another; that means, flip flop and this particular path is basically bypass or if it passes through the buffers in this and this. So, this is now we have to calculate that ok, what is the timing information or how much time I require to calculate the maximum operating frequency of this particular circuit which we have already implemented?

So, this is the; that means, launch edge and this is the capture edge the time between this launch edge to the capture edge is related to or that is the; that means, summation of this delay is the minimum cycle time requirement of your circuit.

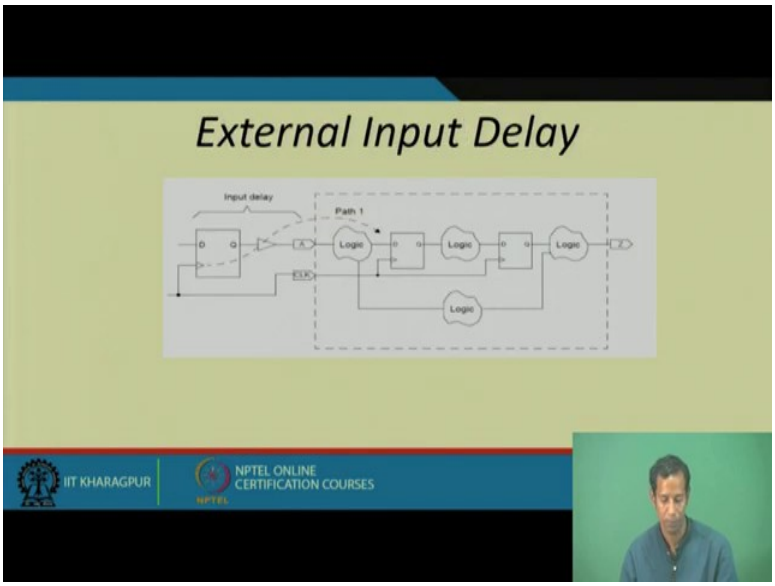
(Refer Slide Time: 15:48)



So, generally if I consider the clock in ideal clock what I said is that, the transition from 0 to 1 and 1 to 0 they are basically instantaneous it means just vertical, but in actual the clock at the register pin is something like this ok. So, this is nothing, but; that means, the ideal one to the real one this is the latency time of the clock which we have already said. And then what I said is that you need sometimes from 10 percent to the 90 percent that is the rise time and from 90 percent to the 10 percent, which is the fall time ok. And then it is not that every clock edge basically occurs at the same.

So, it needs or if that is basically the difference between the clock edges of different pulse clock pulse is known as the uncertainty ok. So, this uncertainty value is nothing but the information about the how much jitter you have in your clock. So, you have to set the uncertainty in at the clock period. So, that it also in real time you will face this uncertainty or your clock should be work as this particular this is a real time clock which will be work inside of your chip. So, that is why you have to consider this uncertainty value, while you are restricting or you are defining your clock.

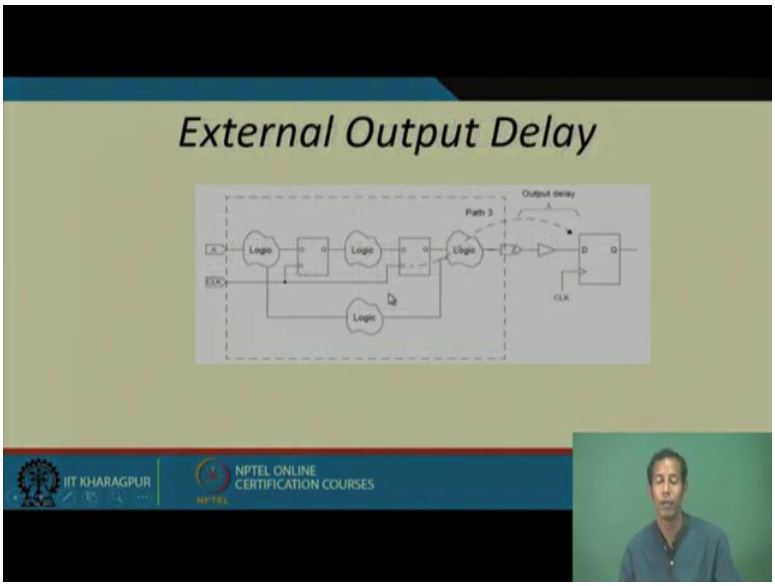
(Refer Slide Time: 17:24)



So, then you we have already seen this we have to consider the input external input delay. So, where from this external suppose this is this, but; that means, this dotted color once this this is my design. And then we are having some of the signals which is coming from the external path so, the external to my chip. So, at that time you have to say it what is the corresponding value which is coming from its input; that means, the clock input to the corresponding input to the first flip flop of your particular chip ok.

So, that much that the information timing information of this particular path you have to set as the external input with respect to this particular clock.

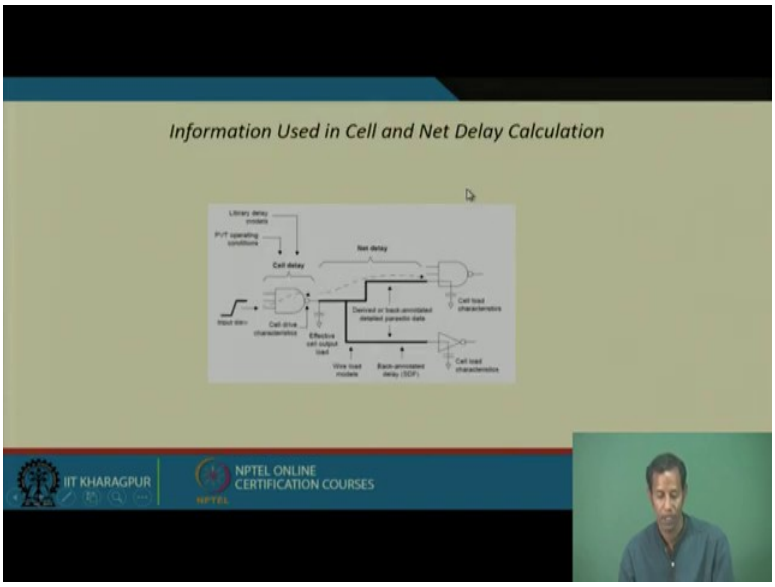
(Refer Slide Time: 18:23)



Then again you have to set the external output delay. So, the external output delay will be the clock pin of your final registers to the clock pin of the; that means, the output side at the output side, what flip flop you have used that means the first flip flop to its where it is connected ok.

So, that will be mentioned the time requirement in this particular path is mean, it should be mentioned as external output delay ok.

(Refer Slide Time: 18:57)



So, then actually we have already know that this gate delays they are basically stored in a look up tables whenever we do this timing; that means, timing analysis. But this net delay or this wire delay they are basically calculated on the fly. So, based on what information they are calculated on the fly is that, suppose you are having this NAND gate and then it is connected with some inverters and it is again connected with some of this NAND gate against. So, these particular gates will be having the cell delay.

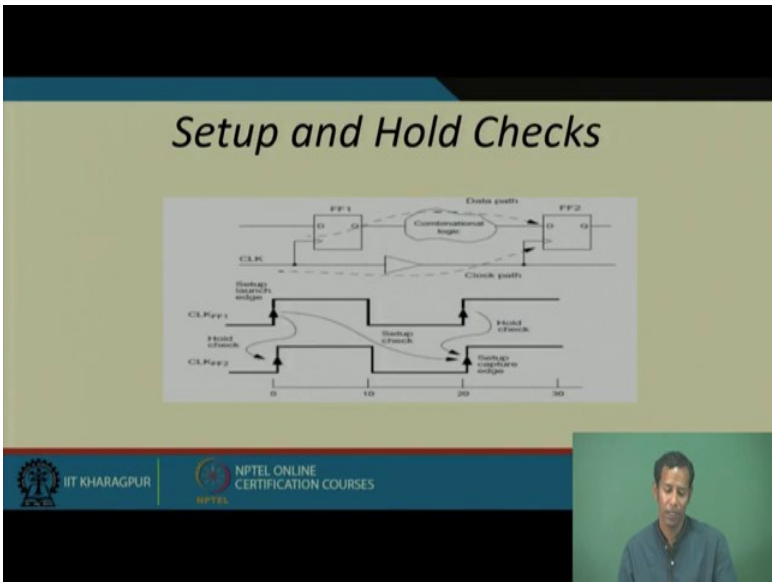
So, cell delay is depends on the process voltage temperature variations as well as the library delay models. So, depending on these two information, this every of this cell delay is mentioned in the library ok. And it also depends that what is the driving that the fan out of this particular gate. If you are using that the fan out of 4 or if you are using fan out of 2 then the delay information will be different for the same NAND gate ok.

Then you are having because of this effective cell output load this will be the delay of this will be different. Then actually as it is connected to this and as it is connected to this. So, this particular this black color one will associate with this net delay. So, how to calculate this net delay? This net delay is basically dependent on the wire load models and the back annotated delays as well as this detailed parasitic data.

So, detailed parasitic data means, depending on the wire load model you will find depending on the length and which metal we are using you are using you will getting the different type of C capacitance and the resistance value. And that means, depending on this different capacitance and resistance value you will find out its delay.

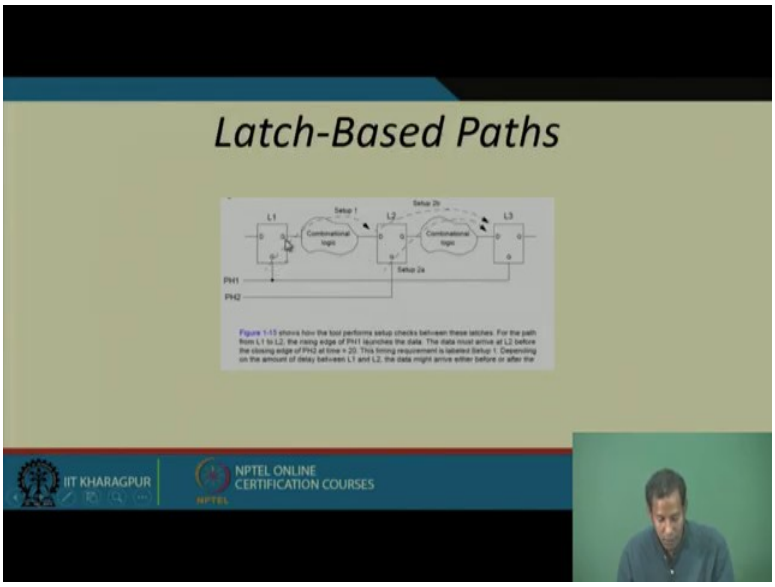
Then again you can calculate for each of this then at that time in timing analysis each of these parts will be just act as a node information. And, this intermediate what is the delay information? They will be just assigned as a weight to the each of this edge and after that it calculates the timing analysis of your full circuit ok.

(Refer Slide Time: 21:34)



So, this is the setup and hold check at this particular edge, you have to consider thus; that means, hold check as well as; that means, for the setup check for both flip flop 1 and flip flop 2.

(Refer Slide Time: 21:53)

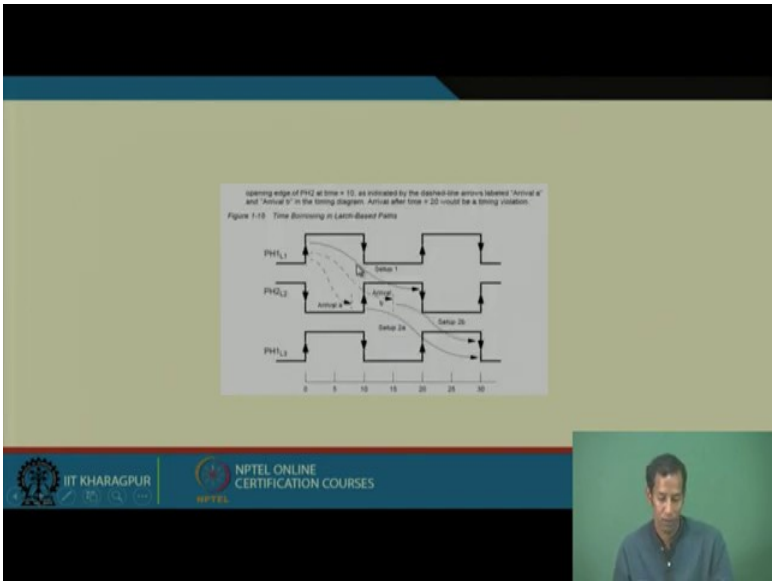


And then again, in latch based design means what actually the primary difference between this flip flop and latch is that flip flops are basically their edge triggered ok. So that means, whenever this edge occurs of the clock then only you will; that means, the operation at the flip flop will occur, but in case of latch it is as it is level sensitive.

So, for the time period the clock is 1 the latch is transparent and; that means, whatever the data input that will be passed to the output. But whenever it is in the 0, the clock is 0 at that time it will be in the; that means, it will be just act as a storage mode. That means, whatever the value is it will be in the latch mode; means whatever the last value it contains it will store that particular information only, it will not take any of the input and it will pass to the output ok. So, the same circuit or the same configuration of the circuit if we consider for a latch based design.

So, at that time what will happen so, here you see we are having this latch based design these are basically replaced with the latch and inside we are again we are having this combinational path. So, for the path from L 1 to L 2 the rising edge of the PH 1 launches the data and the data must be arrived at a L 2 before the closing edge of the PH 2 at time 20. This timing requirement is labeled setup 1 and depending on the amount of delay between L 1 and L 2 the data might arrive either after or before the clock edge occurs ok.

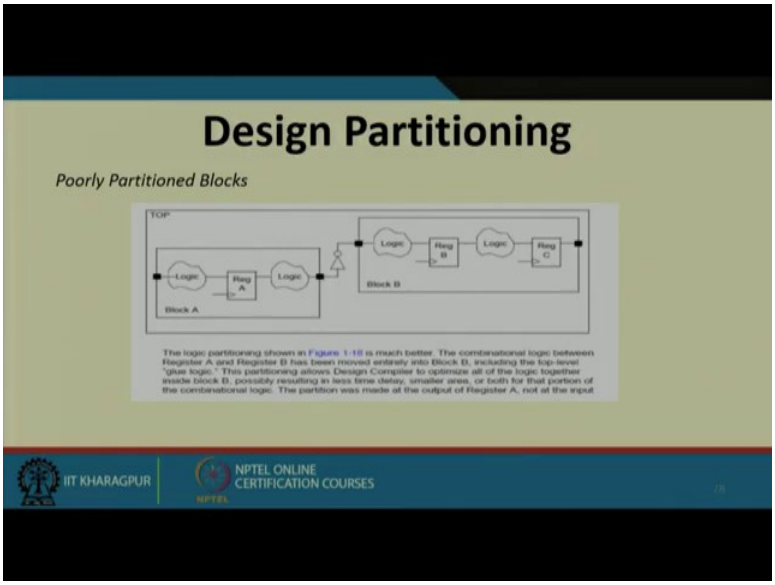
(Refer Slide Time: 23:51)



So, this is the particular check if we do so, here we are having two different clocks; one is this PH 1 another one is this PH 2 and this is PH 1 this is PH 2. PH 2 is just the inverted clock of this; that means, PH 1 and PH 2 they are just inverted complemented to each other. And, then what is this PH 1 L 3 that is the same thing; then we have to check that ok. When this particular output will be arrived at this and when this we have to that

means, for each of this edge of this clocks we have to check the setup timing as well as the hold time, which we have already as seen earlier ok.

(Refer Slide Time: 24:44)



Now another; that means, important aspects of this timing analysis is that this design partitioning ok. So, what is design partitioning is that, suppose actually whenever we are designing one complex circuit. So, at that time you may be having that ok, if I consider one let us say one transceiver system design ok.

So, at that time you might be having this FFT block that encoder decoder block then there might be some filter block, then there might be some other blocks ok. So, it is not that the filters or this FFT or the corresponding that that synchronizer or the source encoder decoder block; every of these blocks they are basically. Like block wise if I consider them so, at that time they will be interconnected they will be interconnected, but whenever we will place them inside of the core.

So, at that time the FFT blocks which the cell requirement will be placed together closer to each other. Why? Because if I place; that means, in one corner one cell of the FFT and another corner of the; that means, in another corner the other cell of the FFT so, at that time the net delay will become too much long and it will take unnecessarily it will increase the cycle time or the critical path.

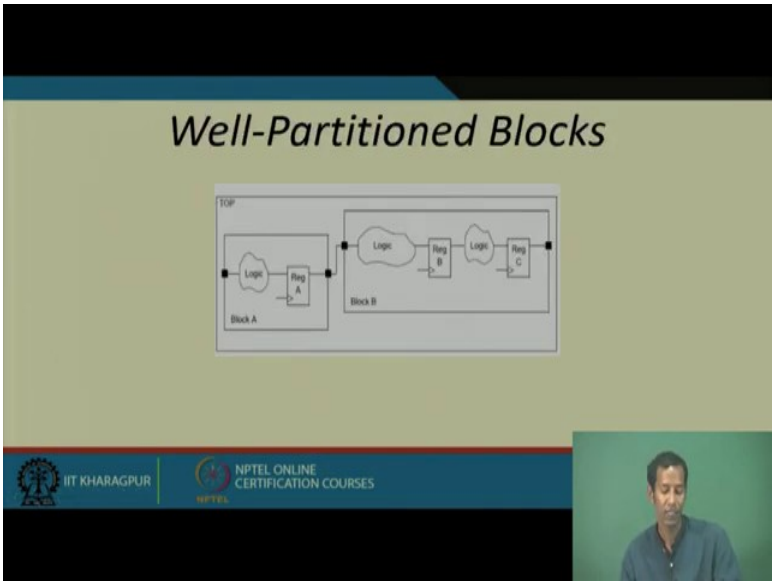
So, that is why always whenever we will place the design so, that we have to remember or the tools has to be remember or we have to put or we have to tell that tool. So, that it can place the design closer where they are having the interconnected connectivity or interdependent connection between the two blocks two sub blocks or two blocks.

So, that is why if you actually if you see that ok, this is block A and this is block B, which is basically part of the main module which is the top. So, at that time you see the logic or from this the output of this block A is the combinational logic. Then again it passes through one inverter and then again it goes to logic of block B and then the reg then the logic and the reg C.

So, whenever we will connect this block A and block B so, unnecessarily this particular will be at that time being the critical path. If I consider the critical path for this this will be only just this logic; the critical path for block B that is the; that means, this logic. Now whenever we will combine this block A and block B at that time it will be 2 t logic plus this inverter delay.

So, this is poorly partition block. So, then this particular partition in this particular block or this particular consideration design consideration, if we change the partition in a different way so, at that time can I improve the performance?

(Refer Slide Time: 28:13)



So, at that time what I do is that, I put the corresponding block A; that means, all this logic I just place to block B ok. And, I am having only logic and registers A over here and I am having the logic and then all this part over here.

So, at that time this is well partitioned block because, I do not need any of this; that means, here also it is after this registers it is going to the corresponding logic. So, all these timing information timing checks will be done perfectly at this and all the timing checks performed here that would be done correctly. So, that is why this timing check for this block A and block B, there will be they will be independent to each other and they will be much more correct much more accurate.

So, that we can get the actually information about the timing analysis report. So, that is why this one as a designer you have to be whenever you are doing this partitioning. So, at that time you have to be more careful how you in a well mannered way you can partition the design. So, that it does not affect the timing analysis further ok. So, whenever we do we have to consider all these things in mind and we have to carry on our digital integrated circuit design ok.

So, with that for today this is it; next day again we will see or we will continue with the topic.