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

**Lecture – 54**
**Timing Issues in Digital IC Design**

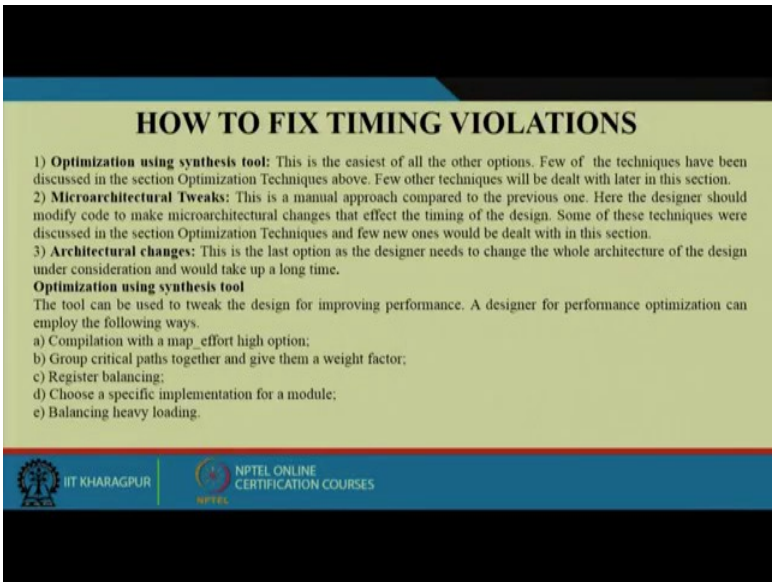
Hello everyone, welcome to the course on Architectural Design of ICs. So, in the last class we have seen the timing; that means static timing analysis basics. So, today we will discuss the Timing Issues in Digital IC Design and why we have to consider this timing issues; mainly if we need; that means, the closer to the actual performance timing information about the; that means, real time integrated circuit which is comes in the form of; that means, physical integrated circuits chips.

So, if we need that to check the performance of when that is fabricated; and we got the black tiny chips. So, how it works, whether all the paths or all these timing information's are correct whenever; because this once we develop this integrated circuit it really; that means, it consists with so many that means, other costs ok. So, once if we have done that so, then there is no such options that ok, we will; that means, remove the error and then again we will redo the things or reprogram the things. So, that every timing information's are that correct.

So, that the digital chips are working fine. So, to avoid that as in digital integrated circuits there is no such options for the errors. So, that is why whenever we are doing this analysis so, at that time we must have to be very much sure that everything is correct or every path is there is no such violation whether there is set up time violation or hold time violation. So, every path meets the timing constraint as well as; that means, all the other constraint which we have already put.

So, that whenever it is working in real time, it just actually performs as what we have thought whenever we are in the designing it. So, for that reason we have to put the constraint to the clock whenever we are designing these digital IC design. So, what are there, what we should put and why we should put those constraints what is the meaning of those constant that we will see in this particular lecture.

(Refer Slide Time: 02:35)




So, we know that in in in actually in timing analysis we can get two types of timing violations. So, what are those one of these timing violations actually will be the set up time violation and another timing violation will be the hold time violation. So, now, the question is that how to if I got the violation. So, at that time how to fix these timing violations?

So, one of the technique is that optimize using the synthesis tool. So, optimize using the synthesis tool means is this is the easiest of all the other options, few of the techniques have been discussed in the sections of optimization. That means, whenever we have already that architectural optimization whenever we have already talked about and few other techniques will be dealt with later in this particular section.

So, mainly this, whatever we have discussed that architectural design of IC which is the name of this course. There what we are trying to do we have tried to optimize the circuit in terms of that mean architectural level. So, we can optimize the circuit in terms of logic levels or in terms of the gate levels. So, other optimization options are also there. So, the timing; that means, if we optimize the circuit then; obviously, there will be improvement in the timing performance of the particular circuit. So, by that also we can fix the timing violation if we are getting any.

So, that is why this is the easiest one; then we are getting; that means, the second option is that micro-architectural tweaks. So, this is a manual approach compared to the

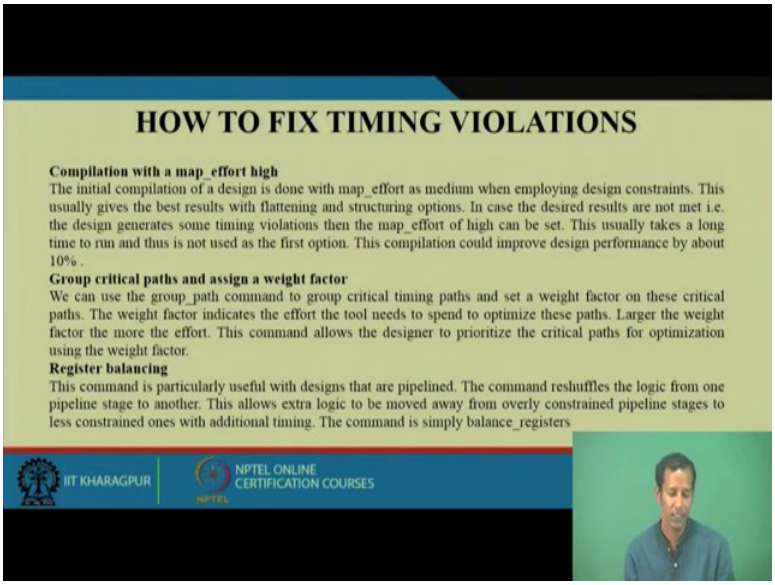
previous one here the designer should modify the code to make micro-architectural changes that effect the timing of the design some of these techniques were discussed in the previous lectures and few ones will be dealt with later in this particular sections and as well as you know this micro-architectural changes.

So, that is why we are basically we are dealing with all these architectural aspects where we can improve the performance in terms of area or whether we can improve the performance in terms of timing ok. So, this the synthesis tool actually what it does is it does the optimization automatically. So, automatically means does it means that it is. So, much intelligent automatically it can do the corresponding optimization, no.

Basically there are some algorithms which is which are running behind of it. So, depending on that, so, who have developed that EDA tool or the synthesis tool they have put those options. So, if you put the constraint to the tool that ok. So, this is my area is my major constraint, or power is my major constraint, or this speed is my major constraint. So, depending on that your constraint which constant you are putting you are telling the tool. So, depending on that corresponding algorithm will be selected.

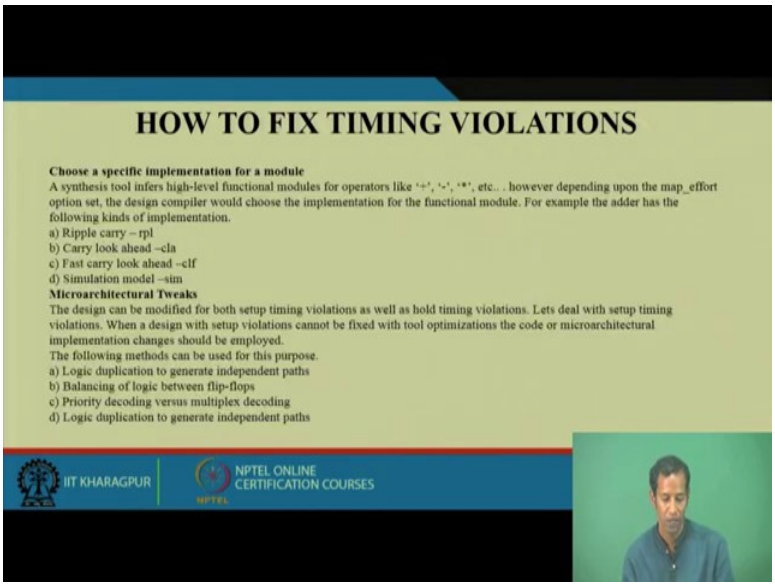
And those algorithms will be applied to your particular code. So, that you can get the best performance by the tool automatically ok

(Refer Slide Time: 05:50)



So, actually these are the techniques by which you can fix the timing violation. So, in in compilation mode you can just; that means, if you can put this map effort high; and then you have this you can group the critical; that means, you can identify the critical path and then we can you can group them and weight; that means, assigned different weight factor. So, that the tool automatically try to modify those paths to get a good performance and then you have other options like register balancing.

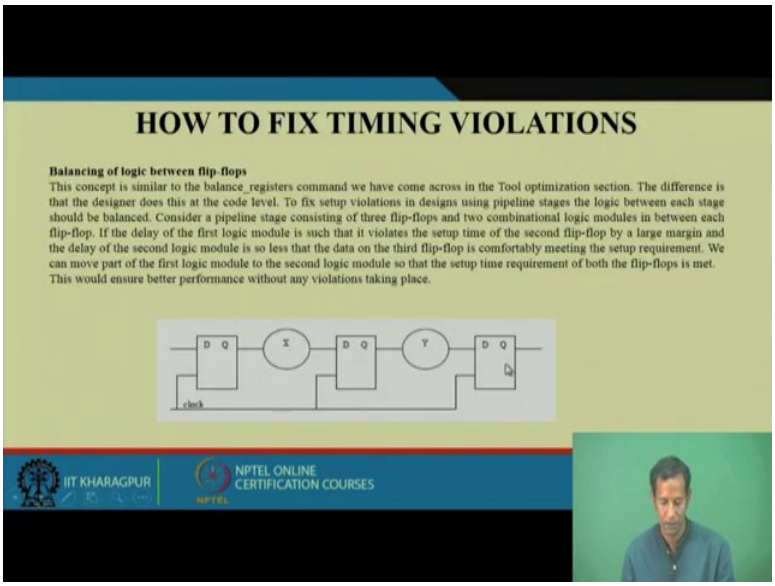
(Refer Slide Time: 06:26)




And then there you are having this you have to choose a specific implementation for a module like if you are a like if you are having this this adder architecture or multiplier architecture or subtractor architecture or divider architecture.

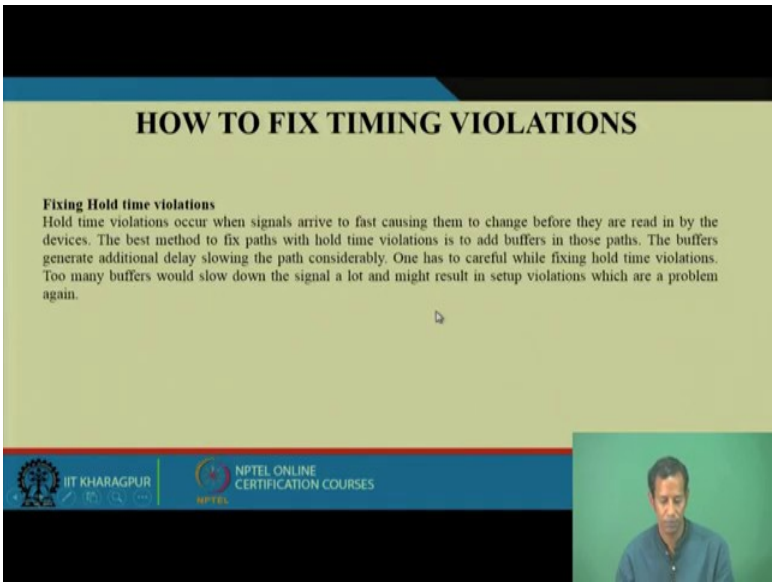
So, those basically try to modify or those will try to optimize this circuit in terms of speed power and area which we have seen till now in the particular course.

(Refer Slide Time: 07:12)




And, so these are the other options by which we can.

(Refer Slide Time: 07:16)




So, now how we can fix the hold violation, ok. So, the hold violation for fixing the hold violation first when the hold violation occurs; that means, when the data arrives too fast too fast means, actually we have already seen when this hold time violation can occur.

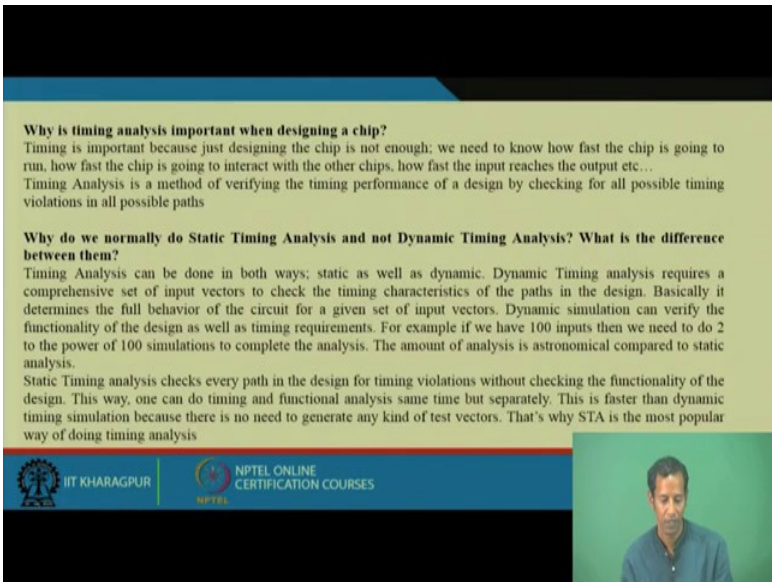
Again we will see when this hold violation can occur, but the thing is that the best method to fix this path is to this add the buffers on the particular path. So, that the data comes or the data; that means, is delayed. Hold violation basically comes because of it

comes it reaches the data too fast. So, if you want to make the data very slow. So, at that time you pour the extra buffers on to that particular path.

So, that if the data now comes at late and you can fix the hold violation ok. So, what is a; that means, this necessity of these buffers? These buffers basically generate as additional delay showing the path considerably and one has to be very careful while fixing the hold time violation. Because, whenever you are putting hold time violation so fixing adding these buffers means you are basically again effecting the clock time requirement the cycle time at that time it will be more.

So, you have to be more careful whenever you are doing or you are applying this techniques and too many buffers would slow down the signals a lot and might result in setup violation which will be a problem again.

(Refer Slide Time: 09:00)




Then why is this timing analysis is very much important while you are designing a chip? Timing is very much important because just designing the chip is not enough we need to know how fast the chip is going to run and how fast the chip is going to interact with the other chips.

How fast the input reaches to the output etc. Because whenever we have designed this digital chip; so at the time you have to place it somewhere. So, that it and it has to be connected with the real time inputs and the real time outputs. So, for that reason what it

is actual performance that we have to know from the beginning while we are designing it is not that the simulation and the actual performance there is a difference.

So, there is a gap. So, if you can; that means, close that gap. So, your simulation results and your simulation result means virtually you are basically you are trying to check the performance. So, when this virtual results and real time results they are much closer to each other. So, at that time you can say that you are; that means, you have designed or you have developed one circuit which is perfectly fine or I can assure you that it will work perfectly fine whenever it will be; that means, place in a printed circuit board or it will be placed for a or it will be used for a real time application.

Then why do we normally do static timing analysis and not dynamic timing analysis, what is the difference between them. So, what is dynamic timing analysis and what is static timing analysis that we have already discussed that in dynamic timing analysis we put the simulation; that means, these test vectors we put and then we check every corners of that. So, that depending on the; that means, signals level.

That means whether that is 0 or 1 depending on that the actual circuit performance also we can analyze that if this signal is 0 then it is working like this, the delay is something like this, or this; that means, the in a delay is this. So, that we can also do in timing analysis which is the example of dynamic timing analysis, but in case of static timing analysis all the information are fixed depending on the which technology library you are using like this gate delay they are fixed this net delay they will be calculated on the fly depending on the wire load model.

Wire load model means where this resistance and capacitance values are stored. So, depending on that how what is the length only that information is required and which metal they you are using only that information is required and depending on that you can calculate the corresponding net delay on the fly and you get the static timing analysis values. So, the timing analysis can be done in both ways static as well as dynamic.

So, dynamic timing analysis requires a comprehensive set of input vectors to check the timing characteristic of the paths in the design. Basically, it determines the full behavior of the circuit for a given set of input vectors dynamic simulation can verify the functionality of the design as well as the timing requirements. For example, if we have

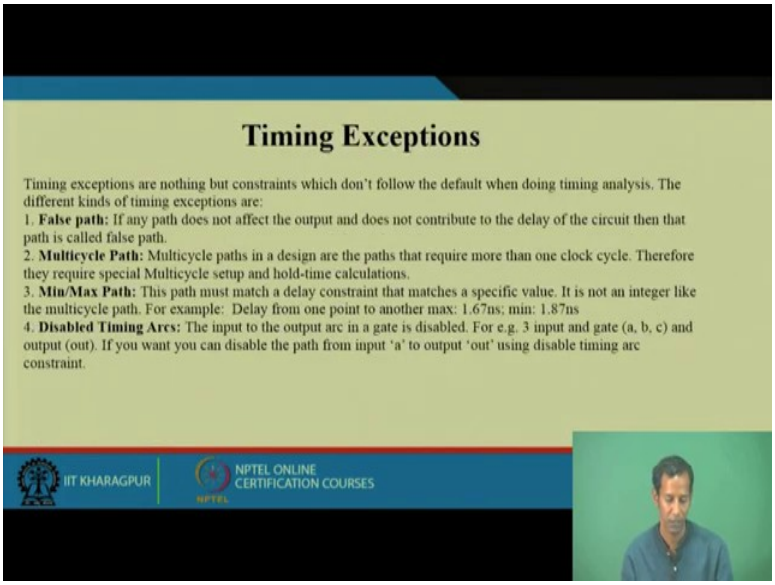
100 inputs then we need to do 2 to the power 2 to the power of 100 simulation to complete the analysis.

And the amount of analysis is astronomically compared to the static analysis. What is the meaning of this is that whenever we analyze the timing; that means, timing issues with respect of; that means, giving the test vectors. So, at the time what is the length of the test vectors. So, 2 to the power of that many numbers test vectors we have to do; that means, that that much of case we have to consider analyzing the full performance of the circuit in dynamic timing analysis.

So, more the number of sets or more the number of variables input variables more the number of case you will get in case of dynamic timing analysis, but in case of static timing analysis as we are not considering any of the test vector on that case. So, at the time this static timing analysis check every path in design for timing violation without checking the functionality of the design.

So, this way one can do timing and functional analysis same time, but separately. So, this is faster than dynamic timing analysis because there is no need to generate any kind of test vectors that is why STA is most; more popular way of doing the timing analysis. So, that is why this static timing analysis is much more preferable than the dynamic timing analysis.

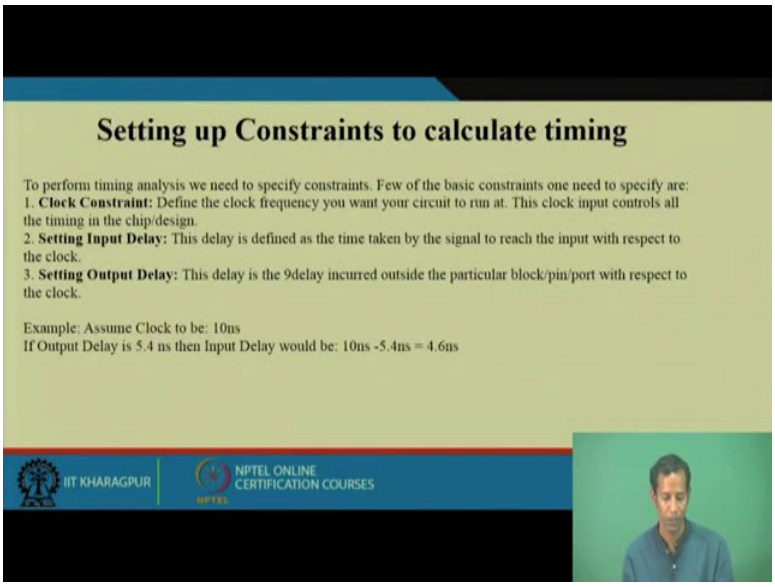
(Refer Slide Time: 13:52)



So, then there are timing exceptions. So, what are those timing exception these timing exceptions are the false path, then the multicycle path, then the minimum or maximum path. Or if you are having one combinational loop so, at the time how you can compute the timing analysis of a combinational loop. So, at the time you have to do this or you have to say or; that means, if you are running the circuit in a combinational loop.

So, combinational loop where you can find like in if you if you have design one registers ok, or if you have design one flip flop, or if you have design one latch. So, at the time you will be find you will be having one combinational loop. So, the timing analysis the static timing analysis of that if you have; that means, if you have to do. So, at the time you have to disable those timing ARCs otherwise you will get the wrong information about the timing in; that means, timing analysis values ok.

(Refer Slide Time: 14:57)




So, what constraint do we put while we do this timing; that means, calculate this timing; that means, timing values or we do this static timing analysis. So, one major thing is that you have to constraint the clock constraint the clock means you have to set what should be the minimum clock cycle time or the clock period of the corresponding design what you are doing.

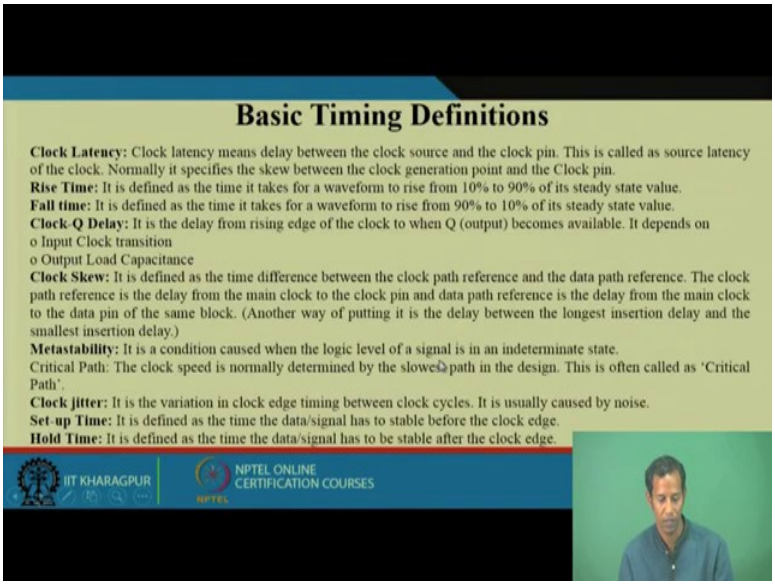
Then you have to do thus this, you have to set the input delay as well as the output delay. So, what is that input delay and output delay is that; that means, it is not that you are having. So, if you consider one; that means the digital chip. So, at that time you must be

having the; that means, the I/O pads they are basically placed little bit far from your core ok. So, whenever this pins which are basically connected to the pads?

So, they need sometimes to reach to the actual pins of your cores ok. So, whenever these times; that means, the time which is required from the input pad to the input pin of your core. So, that information is you can mentioned by putting that is input delay and the same thing on the output side; that means, the output pin of the core to the output pad you can set that as a output delay ok.

So, if we can assume that one clock should be the period of the clock is 10 nanosecond if my input is; that means, if the output delay is 5.4 nanoseconds. So, at that time the input delay should be 10 minus 5.4 that is 4.6 nanoseconds of time ok.

(Refer Slide Time: 16:51)




So, then what are there whenever we are considering any clock. So, at the time there are few things you which you have to know; about which are basically related to the clocks only the clock latency.

So, what is this clock latency? clock latency means the delay between the clock source and the clock pin. So, this is called as source latency of the clock normally it specifies the skew between the clock generation point and the clock pin. And what I said is that, whenever you design one circuit; that means, IC. So, at that time your core has the pin ok. So, that will be having the clock pin.

Now, the full chip will be covered by the that input output pads; where, this pads will be connected to the real world ok. So, those pads are basically there is a gap between those pad pin; that means, the pad cell to your input pin of the core. So, in clock the clock should become externally. So, the clock pad if you are not generating your clock inside of the core. So, at that time the output of the clock; that means the external source of the clock which is coming to the pad then to the pad from the pad to the pin of your core.

So, it needs some times which is again it is connected through wires. So, it will be having some of the delay which is considered as the clock latency. Then rise time; so it is defined as the time to takes for a waveform to rise from 10 percent to the 90 percent of it is steady state value. So, as we know that actually though we have considered the any signals like the transition from 0 to 1 or 1 to 0 there is no such time it takes; that means, the instantaneous we consider in ideal case but in actual whenever we work with the real time signal.

So, at that time it takes sometimes to reach from 0 to 1 or 1 to 0. So, that is why this rise time is basically the time requirement where the waveform takes or the signals basically takes from low to high of 10 percent to 90 percent. So, the time which it takes to go from 10 percent to the 90 percent is known as rise time.

The opposite ones when it comes from 90 percent; that means, from high to low 90 percent to the 10 percent that is the fall time. Then clock to Q delay ok; so, what is this clock to Q delay? It is the delay from the rising edge of the clock to when Q becomes available. So, it depends on the input clock transition as well as the output load capacitance means what, as we have already discussed that the register itself will be having some inherent delay because it is made up made by some of the; that means, gates ok.

So, those gates will be having some of the delay. So, from the clock to the output clock is the input pin. So, from clock to the output; how much time it takes that is the delay of that particular resistor which is known as clock to Q delay and it depends upon the input clock transition as well as the output load capacitance. So, if one particular resistor is connected to several other loads, so, at that time depending on the loads these particular paths will take much time.

So, then again clock skew clock skew we have already defined already we have discussed. Again actually if I say that it is defined as a time difference between the clock path reference and the data path reference the clock path reference is the delay from the main clock to the clock pin, and the data path reference is the delay from the main clock to the data pin of the same block ok.

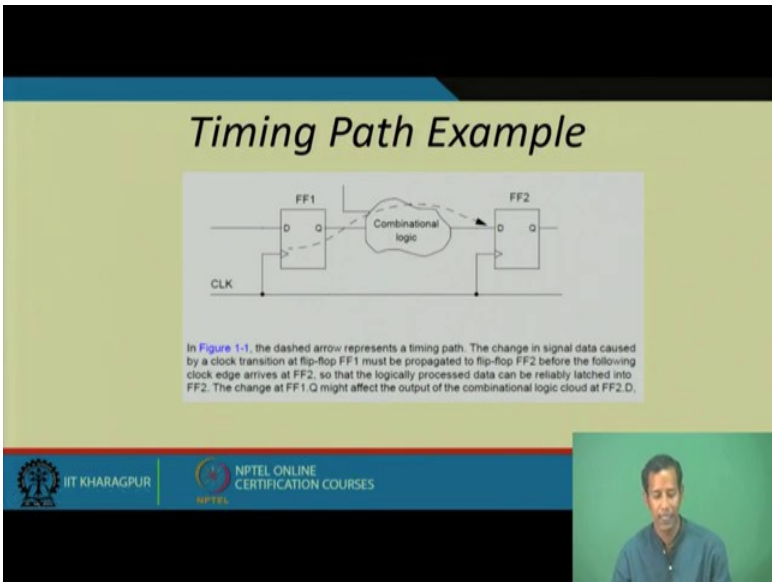
Then metastability; metastability is nothing but a state it is the condition caused when the logic level of a signal is in-determined state or it is in; that means, it is in you cannot recognize the data properly whether that is 0 or 1 unambiguous state, if you find then that state is known as metastable state. Then there is another; that means, point important point which is the clock jitter it is the variation in the clock edge timing between the clock cycle, it is usually caused by the noise.

So, what is this clock jitter clock jitter means, suppose the clock is a continuous source right. So, it is not that every rising edge and the falling edge of the clock; suppose, clock means I am having if I consider one particular clock cycles so; that means, whatever is the time period we set. So, from the rising edge to the falling edge and then again if it finds the rising edge up to that is the clock period.

So, if I consider let us say 100 cycles. So, if I; that means, in superimpose them one after another of these 100 cycles. It is I will not find that every of this rising edge of the clock is same to each other, there will be slight difference in the rising edge as well as the falling edge which is basically caused because of the noise which is associated with this clock pins. So, that is why this basically creates the clock jitter and as I said that clock is the major component in digital circuits.

So, that is why you have to be more careful more cautious while you are designing with or you are playing with the clock in the design. So, then again set up time and hold time this we have already discussed the time requirement before the clock edge arrives when the data must be stable that is the set up time and that that the hold time is the when clock edge arrives then again still we need sometimes to properly recognize the data. So, that time is the hold time ok.

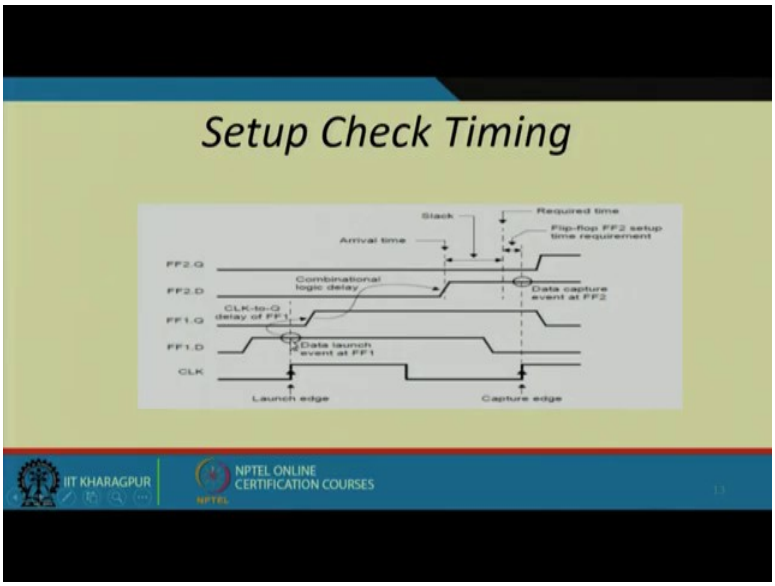
(Refer Slide Time: 23:21)




Now, then again actually if we consider one of this; that means, two flip flops this is the basic consideration of any; that means, detailed circuits that there will be two registers and in between of them there will be the combinational logic which defines; this the maximum critical path of this will be define the clock; that means, the minimum clock period requirement of these two registers ok.

So, apart from that not only this we have seen that the set up plus the skew plus the hold all these basically constraint the minimum requirement of the clock in this kind of circuits ok.

(Refer Slide Time: 24:52)



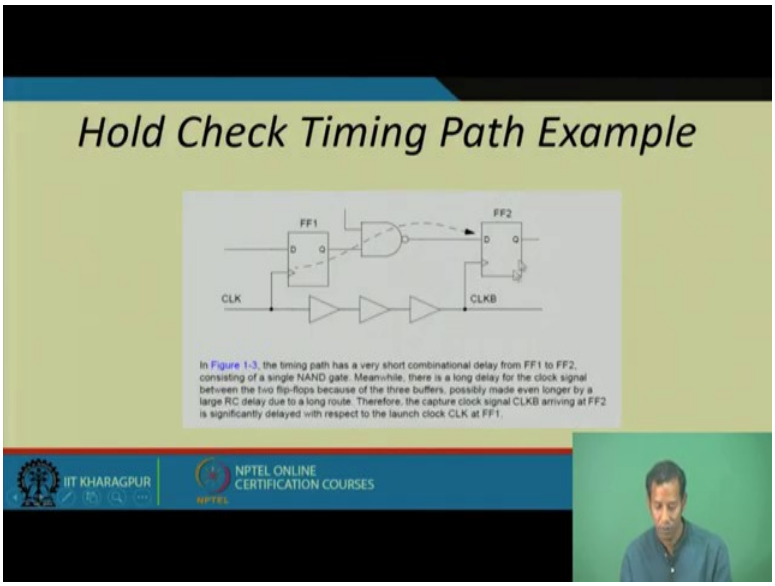

So, now whenever we are actually we are doing suppose this is the case if we consider. So, this is flip flop 1 this is flip flop 2 and in between we are having some combinational logic; and both the clock both the clocks are basically driven by one single clock pin.

So, at the time if we have to do or how we have to do this set up timing check is that this is my clock this is my clock source. So, if these two particular flip flops are positive edge trigger flip flop. So, at that time whenever I find this clock edge and this clock at that time this is the input data of the flip flop 1. So, if it is this is known as launch edge. So, launch edge will find that the data input value is 1 over here.

And after that, I will get the output at Q after some time why after some time because this is the requirement of this clock to Q delay we are having for this flip flop 1 ok. So, then the output of the; that means, flip flop 1 that will be the input to the combinational logic and then again after that it will be the input to the second clock sorry second register.

So, this now will come over here; that means, this will come because of the combinational logic delay if it is arrives here and then again.

(Refer Slide Time: 26:12)




For the next for this particular clock, now what will happen in this edge in this edge both the flip flop will try to find what it is the value of its data. So, in this particular case the data for flip flop 1 is 1.

So, that is why this flip flop ones output comes over here, and as this Q is 0 over here and this edge for the this. So, at the time it will consider to continue to 0. So, this will come at this particular time then again at the next clock edge it finds that this flip flop ones data is 0.

So, again it will be after sometimes it will come down to 0 and here at this particular case the input of the flip flop 2 is 1. So, again it will be goes to 1 after some time. So, this particular requirement these delay is coming because of the clock to Q delay ok. So, then this is the actual arrival time and this is the required time of the data which will be coming.

So, the slack amount is the required time minus the actual arrival time is if it is positive. So, at that time I can say that the design will work perfectly fine ok. So, this is the how actually in circuit or in static timing analysis we do the setup timing check or the tools actually not ask the tool automatically performs the setup timing check while it does the static timing analysis in the circuit ok.

So, this is the basics how these tool works when it runs the static timing analysis on the corresponding digital circuit what we have designed. So, for today this is it next day again we will consider more on these particular aspects.

Thank you.