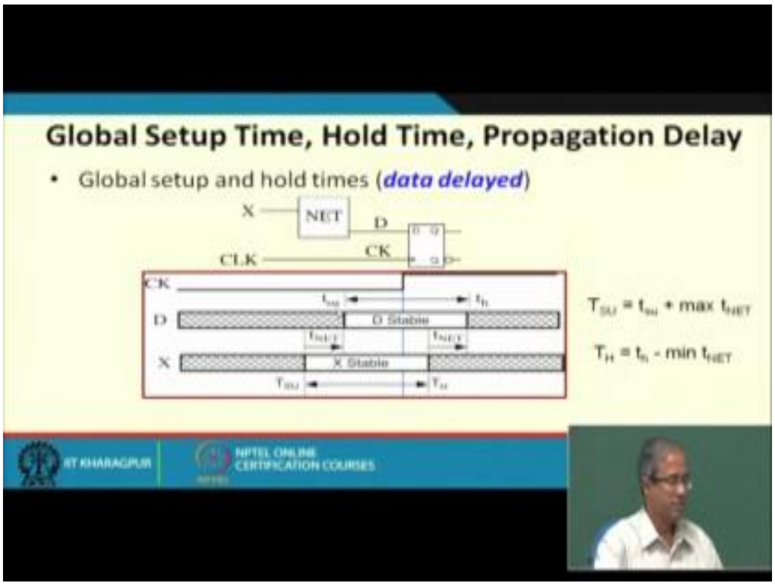
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 26
Clock Design (Part III)

So, to continue the discussion we were discussing various scenarios when one flip flop is driving another flip flop through some intermediate combination circuitry, that how this setup and hold time and other delays propagation delays and other delays, they affect the maximum frequency of operation of the circuit.

(Refer Slide Time: 00:48)



Global Setup Time, Hold Time, Propagation Delay

- Global setup and hold times (***data delayed***)

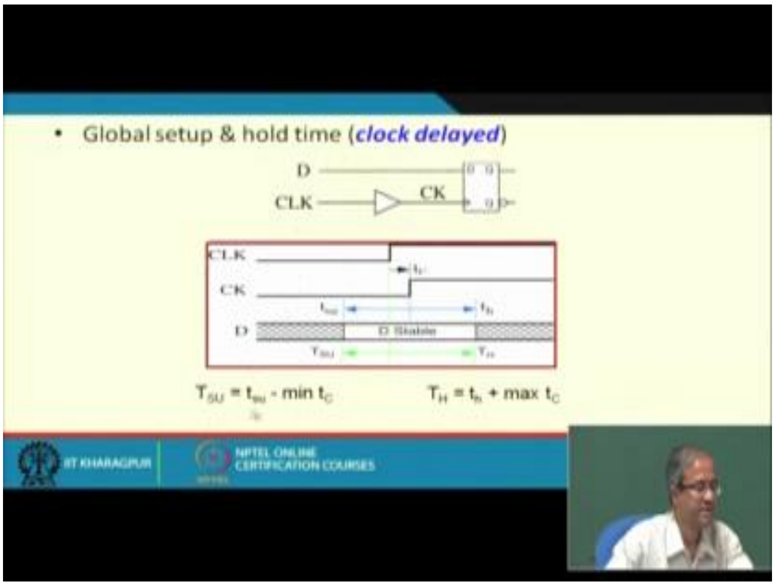
$T_{SU} = t_{su} + \max t_{NET}$

$T_H = t_h - \min t_{NET}$

So, we continue our discussion in that line. So, we look at global setup time hold time propagation delay to start with. You see here we consider a scenario where data is delayed; that means, I have a flip flop I have a combination circuit, some inputs are applied I am treating it in a global sense not one flip flop driving another flip flop. Now due to some reason X is getting delayed. So, let us see this clock signal is like this, so what are the requirements? The D input is coming to the flip flop. So, before this, this T setup constraint must be satisfied and this T hold constraint also must be satisfied. So, X well assuming that the delay of this network is 0, so the data X must also be stable from here up to here, but because of the delay of this network it has to be stable a little earlier T net. So, this is the ideal case where T net is 0, but when there is a finite delay of the network. So, we need D to be stable here T setup time before the clock, but because of the delay of this network my X has to be stable at least this much time before that, and it should remain stable at least up to this much because after that the propagation delay will take care of it that data will reach here after T H.

So, this is the constraint that needs to be satisfied here. So, T s u as you can see T s u is this, plus maximum of T net. T s u plus T net that should that will be your T setup time setup time of this global input. So, the input setup time will be the delay of the network maximum, maximum delay plus setup time of the flip flop and the hold time of this global input T H will be T H the hold time of the flip flop minus the maximum you know this will be minimum delay, because it is going in the negative direction you have to take the minimum delay minimum delay of this network. So, from this diagram these two equations are pretty clear, at how these are coming. So, when you are feeding a data. So, your data must be applied and kept stable at times, which are governed by these two expressions.

(Refer Slide Time: 03:36)

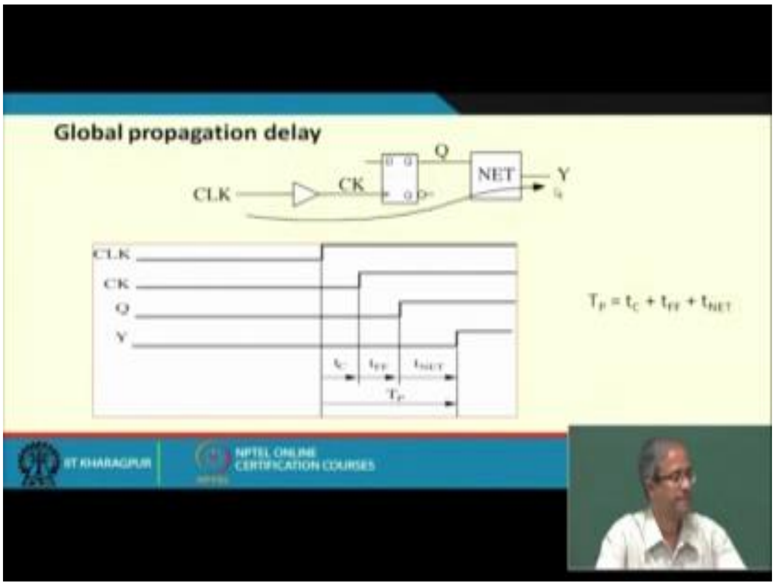


- Global setup & hold time (***clock delayed***)

$T_{SU} = t_{su} - \min t_C$ $T_H = t_h + \max t_C$

Now, if clock is delayed. So, global setup and hold time is the single flip flop again, now clock is getting delayed there is a delay element it is getting delayed. So, your CLK input clock is this, this is without any delay this CK is like this. So, in ideal situation you must have this D input coming, setup time before that hold time before that. So, for the input also D also, this will be the setup time and hold time. But if this is delayed this edge will get on to the right, so it will become like this there will be a delay of T c. So, what will happen, this D will get pushed here. So, the constraint will be T s u will be small t s u minus T c. This will be T s u clock edge minus this minimum and anything goes in the minus direction you will have to take minimum, anything going in the plus direction you will have to take maximum; because for a general combination circuit the delays can vary minimum path maximum path and for the hold time in a similar way, it will be T H plus t c and because it is plus you take maximum t c. So, for scenario like this where the clock is getting delayed, you have to make calculations like this and apply the external input in this fashion.

(Refer Slide Time: 05:22)

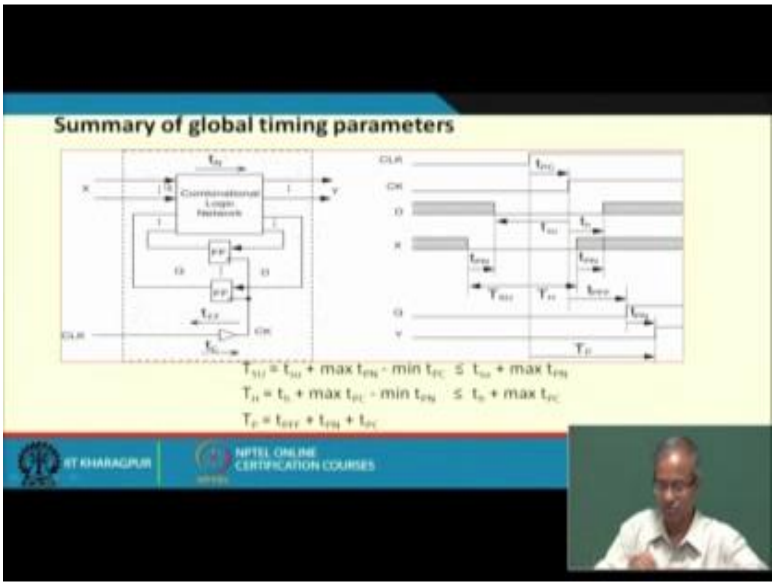


Global propagation delay

$T_P = t_C + t_{FF} + t_{NET}$

Now, global propagation delay again say from a circuit like this, if there is a network which is driving you have to take the output like this then you have to sum up the delays of the all sub components. Like t c will be the delay of this clock, t f f will be the delay of this flip flop, t net will be the delay of this network. So, the whole propagation delay will be sum of these 3 this is simple. So, when you have a path like this. So, see if your clock is delayed your everything is getting delayed, so unless the clock comes you cannot start the calculation. So, your total propagation delay will also include that clock delay. So, only after clock delay things will start and we will have to consider the other delays fine.

(Refer Slide Time: 06:14)



Summary of global timing parameters

$T_{SU} = t_{su} + \max t_{PN} - \min t_{PC} \leq t_{su} + \max t_{PN}$

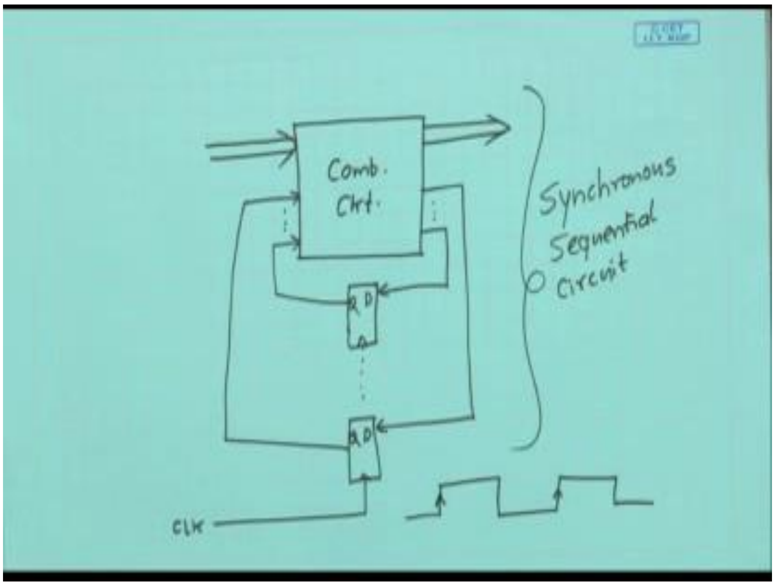
$T_{H} = t_{h} + \max t_{PC} - \min t_{PN} \leq t_{h} + \max t_{PC}$

$T_{P} = t_{PFF} + t_{PN} + t_{PC}$

Combinational Logic Network
X
Y
FF
FF
CLK
CK
CLK
CK
D
X
Q
Y
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, this is to summarize the global timing parameters, you see here I am showing a general model of a sequential circuit. I am just explaining once more before going back to the slide.

(Refer Slide Time: 06:29)



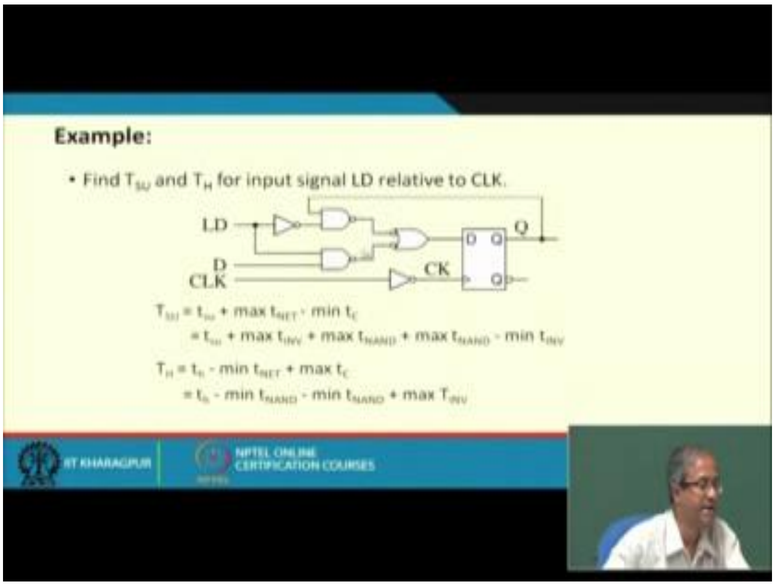
You see any synchronized sequential circuit can be modeled like this; I can model it as a combinational circuit part there will be several flip flops. So, there will be some lines which are primary outputs, some lines which are primary inputs coming from outside, some of the lines will be going to the flip flop inputs like this. And if flip flop outputs they will be feeding this combinational circuit again and there will be a clock that will be feeding all the flip flops together. So, this is how you can model a synchronous sequential circuit. So, every time a clock comes a clock comes like this. So, whenever there is a positive edge of the clock, so whatever is here in the output of the combinational circuit that will be get stored in the flip flop and the new value will come here. So, before the next edge comes, this new value will be used to calculate the new output and when the next edge comes that new output will get stored again.

So, in this way cycle by cycle computation is going on in the sequential circuit. So, this is a model of a synchronous sequential circuit. Now coming back to this diagram let us see, here we have that kind of a combinational circuit and flip flop that are sequential circuit as I said, and we are assuming that the clock that is fed also incurs a delay here there can be a delay T c here, and each of the flip flop is incurring a delay of t f f and this combinational logic network is incurring a delay of t n. So, the timing requirement will be like this in terms of globally you want to see because x is a global input.

So, T setup time like here diagrammatically it is shown here also. So, when the clock comes this CK is getting delayed by t p c it is getting delayed and when this CK comes this is feeding the flip flops. So, the D inputs of the flip flops will incur the setup time, minimum you have to be this is the edge so this is the set up time, minimum and after that this is the hold time and t p n is the delay of the combinational network. So, this setup time of a x must be t p n time before that and here also similarly there will be a hold time. So, the setup time will be this t s u, plus the delay of the network minus this delay of this t p c, delay of this clock network. This clock network is going to delay because when clock is getting delayed you subtract and hold time also there will be similar calculation and T p the total propagation delay with sum of the propagate of the flip flop, the combination network and the clock.

So, this summary of whatever we discussed in the in the earlier slides same thing is summarized here. So, if we have a circuit like this. So with respect to the global parameter the input what will be the setup and hold time with respect to the inputs, and how you can calculate that. So, with respect to the clock edge; so how much before you have to apply X and after the clock edge comes minimum how much you have to keep holding the value of X these are important all right.

(Refer Slide Time: 10:57)



Example:

- Find T_{SU} and T_H for input signal LD relative to CLK.

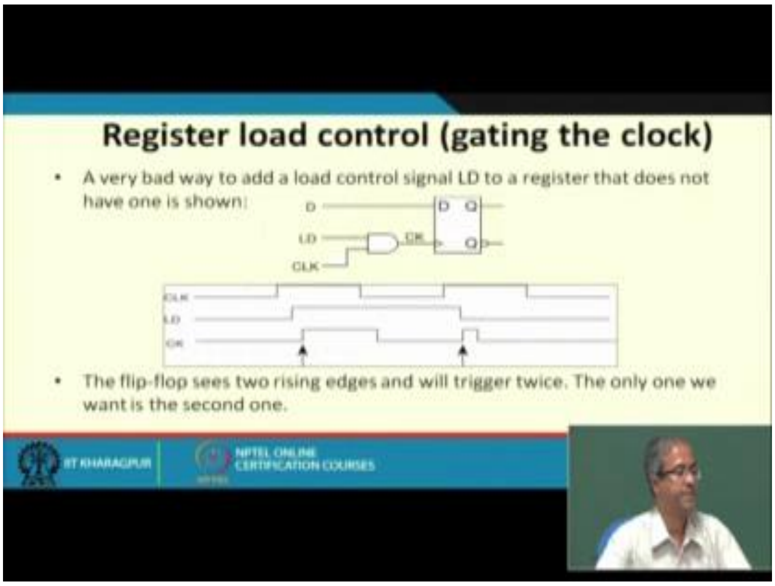
$$T_{SU} = t_{su} + \max t_{NET} - \min t_C$$
$$= t_{su} + \max t_{INV} + \max t_{NAND} + \max t_{NAND} - \min t_{INV}$$

$$T_H = t_h - \min t_{NET} + \max t_C$$
$$= t_h - \min t_{NAND} - \min t_{NAND} + \max T_{INV}$$

Fine here we have an example where we are trying to find out the setup and hold time for the input signal load relative to the clock, these are illustrations of this same case just a specific example. So, here the combinational network consists of 3 levels of gates, one inverter; one stage of NAND, another, this is also NAND gate one stage of NAND. So, the setup time will be, you see the setup time was what in the earlier slide, this was the expression right T setup plus max p n. So, will T setup this one we are using this equation max propagation delay minus delay of the clock. So, max network minus delay of the clock, max network means what max of an inverter, max of a NAND again max of a NAND and clock is only inverter minus min of a inverter; so this is this setup time of the global input load.

Similarly hold time similarly can be calculated this was the expression and min t net will be same thing inverter NAND, NAND, and min minimum delay is what this was the max delay, min delay will come from here this was the max delay not NAND and NAND min delay will come from this path, NAND and NAND. So, only NAND and NAND are there and max this is this.

(Refer Slide Time: 12:43)



Register load control (gating the clock)

- A very bad way to add a load control signal LD to a register that does not have one is shown:

- The flip-flop sees two rising edges and will trigger twice. The only one we want is the second one.

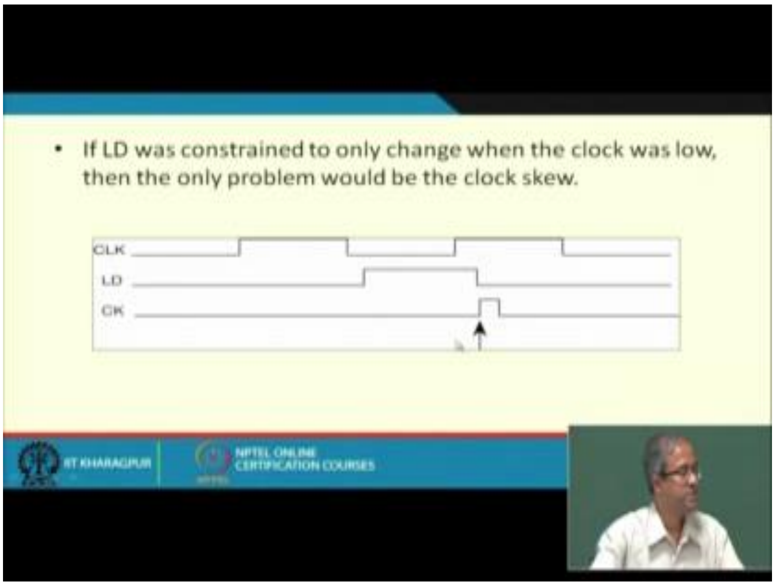
IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

So, you can calculate T s u and T H of the load signal with this expressions right. Now here there is a illustration of a case which is very important, sometimes we want to use a load control of a flip flop like a clock is coming, but not always we want to load the data into the flip flop of the register. So, whenever we want to load we will activate a load signal only then it will load. Some of the register or flip flop can have a separate load control input, which we can use if there, but suppose it does not have it only has D Q and clock.

So, how do you control the load? So, there are few alternatives we are exploring. So, the first alternative is what is saying is that this load signal, we are using an end gate to get it to the clock signal to generate the clock of the D, but what I am say what this is this not a very good idea why? You see the clock signal is coming like this, the load signal will be active sometimes suppose it is high like this, this CK signal for a single clock cycle you see this CK will be active twice, but actually what you want, you want that for one clock period this CK should come only once maybe this second time this one this should not be there. So, if you get it like this depending on when you are making load high and when you are making load low, your number of clocks is generated maybe wrong maybe two times.

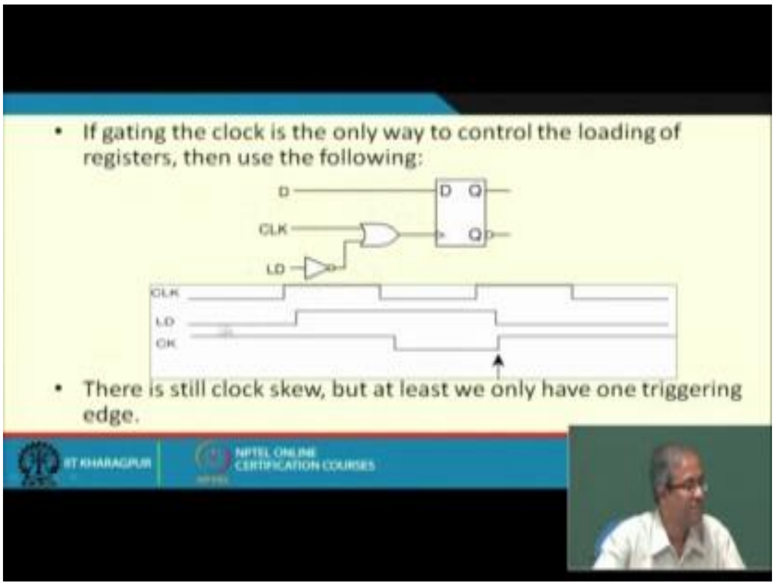
(Refer Slide Time: 14:30)



- If LD was constrained to only change when the clock was low, then the only problem would be the clock skew.

So, there is one solution first we suggest, see here what we are doing we are activating load here when the clock was high we were doing that. Second time what you are doing we are making a constraint, we are restraining load we are saying that we can activate load only when the clock is low; that means, we are activating low only when the clock has reached low, then you see the first pulse will not come only the second pulse will correctly come. But only if there is a clock skew then there will be a problem because these edges can be moving forward and backward because of skew, but otherwise we have solved this problem right.

(Refer Slide Time: 15:17)



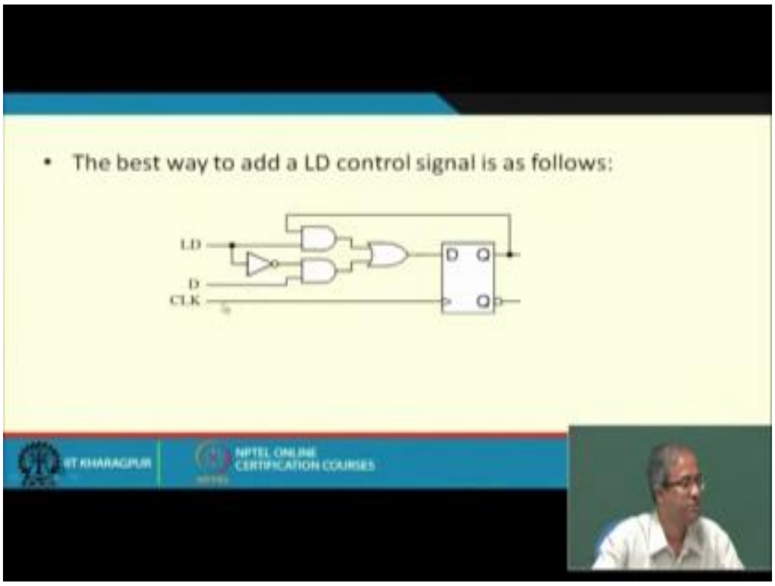
- If gating the clock is the only way to control the loading of registers, then use the following:

- There is still clock skew, but at least we only have one triggering edge.

But again here is another solution which also avoids the problem irrespective of when you are activating the load, that instead of using an end gate you use an odd gate and you invert the load and apply to the odd gate. So, you see if you apply load like this. So, you are making it high when the clock is high itself like in the previous example, and if you calculate the output of this clock it will come like this a single pulse, the first pulse will not come.

So, there can be clock skew, but only one triggering edge per load pulse. So, at least here also we have solved it, but what designer say is that these are you can overcome the problem by doing a lot of tricks like this, but the best way to handle this is to use a circuit like this, because you see to incur in the clock input is highly you can say this is something which you must not do or should not do. Here we are introducing some circuitry to the clock input to feed the clock through some other circuitry, but what people are saying is that this is not something which is good.

(Refer Slide Time: 16:44)

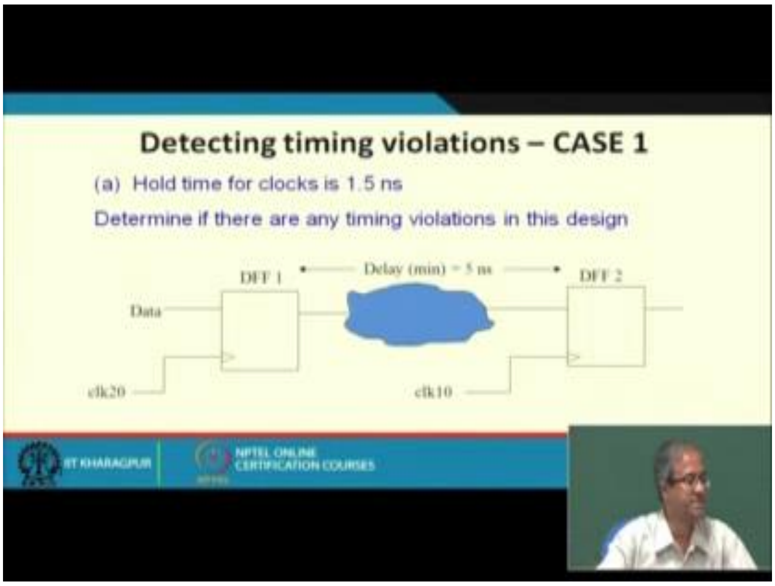


- The best way to add a LD control signal is as follows:

You should not play with the clock, clock should go directly to the clock, but rather you use some extra circuit to handle this problem. So, what is this extra circuitry? This is nothing, but a 2 line to 1 line multiplexer, this multiplexer has 2 inputs one is a D input, other is that Q output that is also coming as a input and load is the select line. If load is 0 then D is selected; that means, you are loading, if load is 1 then Q is selected which means no change. So, whatever are stored that same value will get stored again.

So this is a very safe way of incorporating the load feature in a flip flop, which does not have a load input right.

(Refer Slide Time: 17:38)



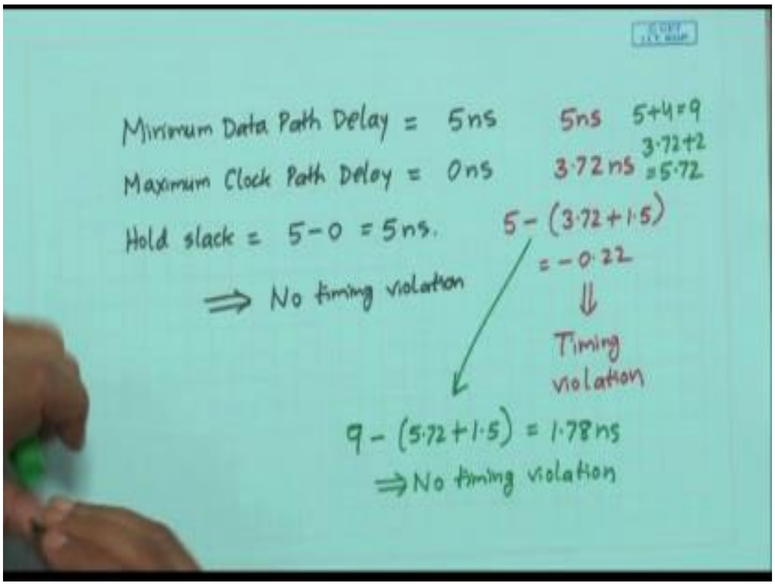
Detecting timing violations – CASE 1

(a) Hold time for clocks is 1.5 ns

Determine if there are any timing violations in this design

So, let us look at a few examples here. So, where we shall just illustrate some timing violations in 3 cases whether there are any. Let us look at a scenario there are 2 flip flops; the first flip flop is feeding to some combinational circuit that is going to the input of another flip flop. There are two separate clock inputs let us call them clk 20 and clk 10, the hold time are 1.5 nanoseconds let us say. So, what we are trying to explore is to determine whether there are any timing violations and delay of this combinational logic is minimum 5 nanoseconds. So, here let us workout.

(Refer Slide Time: 18:33)



Minimum Data Path Delay = 5ns 5ns 5+4=9

Maximum Clock Path Delay = 0ns 3.72ns 3.72+2 =5.72

Hold slack = 5-0 = 5ns. 5 - (3.72+1.5) = -0.22

⇒ No timing violation

⇓

Timing violation

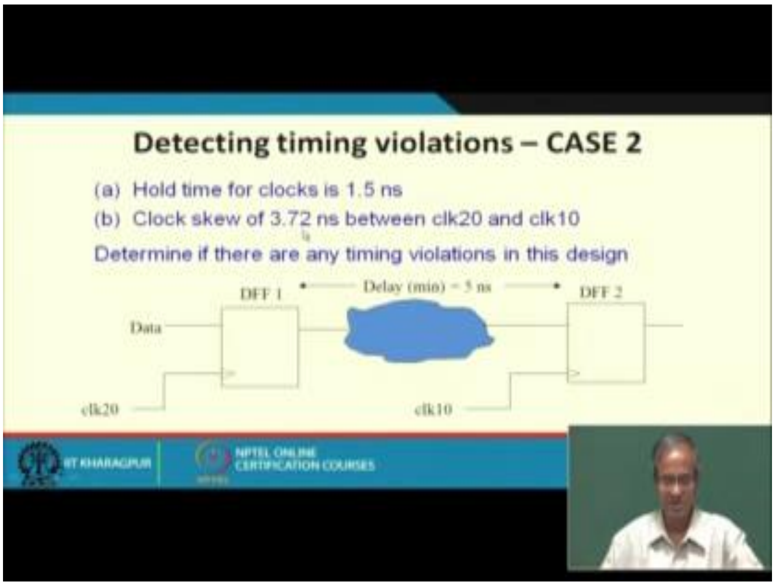
9 - (5.72+1.5) = 1.78ns

⇒ No timing violation

So, the first thing is that let us say; the minimum data path delay in this case 5 nanoseconds maximum, clock path delay we are not assuming any skew so 0 nanoseconds. So, let us call it hold slack is the separation between these two delays. So, how much difference is there? 5 minus 0 this is 5 nanoseconds. Now if this is positive we can imply that there is no timing violation, this you can think that why. So, here we are looking at two things: minimum data path delay and maximum clock path delay look at the circuit once more.

Minimum data path delay is this one data path is this one 5 nanoseconds, clock path delay we are not assuming that there are any skews here now coming together. So, the hold slack will be data path delay minus clock path delay, that is 0 it is because you can compare it with another case let us look another case side by side the case 2.

(Refer Slide Time: 20:16)



Detecting timing violations – CASE 2

(a) Hold time for clocks is 1.5 ns

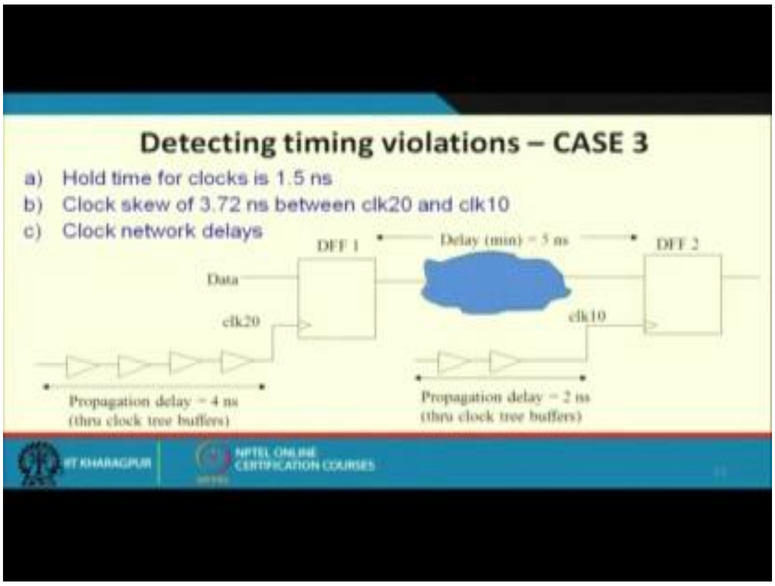
(b) Clock skew of 3.72 ns between clk20 and clk10

Determine if there are any timing violations in this design

Here we are saying that in addition to that there is a clock skew 3.72 nanosecond. So, here what will be the calculation? Side by side we are showing minimum data path delay, so here also will be 5 nanoseconds. Maximum clock path delay because of the clock skew, it will be 3.72 nanosecond. Hold slack you see there is a hold time for clocks, because of the clock skew that hold delay will also come in. So, this will become 5 minus 3.72 plus that hold time 1.5, this will be minus 0.22 this is coming negative which means there is a timing violation

So, in this second case if there is a clock skews of 3.72 nanoseconds, you cannot satisfy the timing requirement anymore, there is a timing violation which is coming in.

(Refer Slide Time: 21:36)



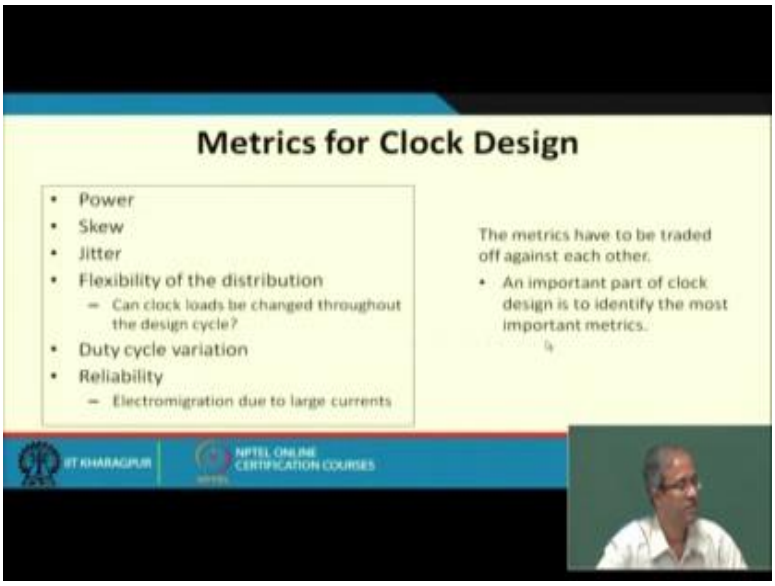
Detecting timing violations – CASE 3

a) Hold time for clocks is 1.5 ns
b) Clock skew of 3.72 ns between clk20 and clk10
c) Clock network delays

So, lastly a third case; so, here we are showing some additional delays in the clock network, suppose there are delay of 4 nanoseconds in the first clock network, delay of 2 nano seconds in the second clock network. So, in this case let us similarly workout the delays; so in the first case it is delayed by 4 nanoseconds. So, it will be 5 plus 4 it becomes 9 nanoseconds. Second one there is a delay of 2 additionally. So, it would be 3.72 plus 2 which is 5.72 nanoseconds. So, this calculation how it will be now, it will be 9 minus 5.72 plus that hold time of 1.5, this comes to 1.78 nanoseconds, because this is positive here also there is no timing violation right. So, this 3 examples actually tell you that depending on the scenario if you are given the data, then you can play around with the data and make simple calculation and compression and find out whether you are violating the timing requirement anywhere or not.

So, you can see clocking is not a very trivial issue, calculating or just evaluating whether for a certain scenario for certain delay scenarios your clocking will give you correct operation, you have to understand the clocking issues very clearly and we have to check whether all the constraints that are supposed to be satisfied they are getting satisfied or some of them are getting violated. So, these are something that you need to do.

(Refer Slide Time: 23:48)



Metrics for Clock Design

- Power
- Skew
- Jitter
- Flexibility of the distribution
  - Can clock loads be changed throughout the design cycle?
- Duty cycle variation
- Reliability
  - Electromigration due to large currents

The metrics have to be traded off against each other.

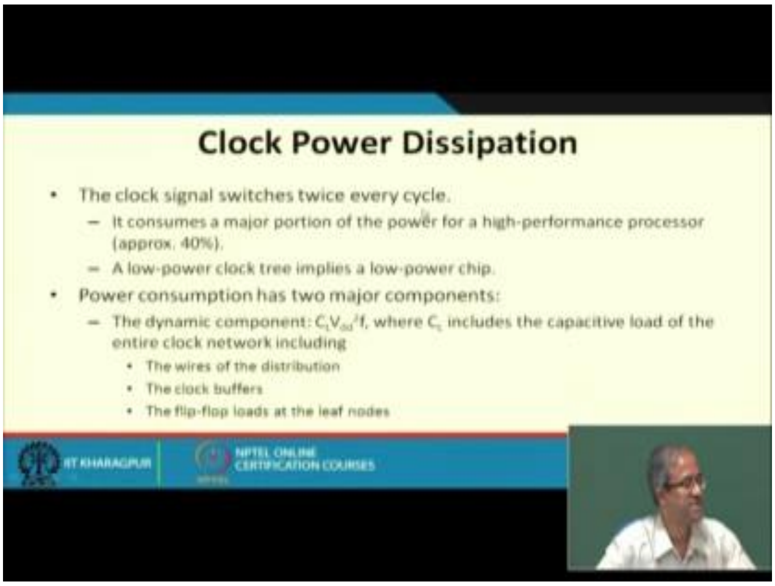
- An important part of clock design is to identify the most important metrics.

Then coming to the metrics for clock design; so when you are designing a clock circuitry, there are lot of issues that you need to look at (Refer Time: 24:02) some of this will be addressing later in our later lectures of course, the total power requirement. You see clock is a network which is heavily active, there is a signal transition at every cycle and for any cmos circuitry you know whenever there is a signal transition there is dynamic power dissipation. So, clock power dissipation is one of the heaviest power (Refer Time: 24:27) the networks inside the chip. Skew jitter, maximum skew maximum jitter maybe some matrix that you may want to just enforce.

Flexibility of the distribution says can you change can you add additional clock leaf nodes, which means you are changing the clock loads means at anytime in the design cycle, because it means to happen that if you want to make some modification some skew values are changing drastically. So, you may have to redo or re redesign the whole network, it should not be like that flexibility should be there.

So, how much duty cycle variation it can tolerate and of course, reliability; because of large current flows there should not be some variation in the physical properties due to electro migration which might lead to wrong operation. So, most of this metrics are mutually conflicting, they have they have to traded off and of course, for a particular scenario we have to look at which one is the most important issue for you. For some application power can be the most important issue, for high performance circuits skew and jitter can be very important, for programmable devices maybe flexibility of distribution are important. So, there can be mutually conflicting issues.

(Refer Slide Time: 25:57)

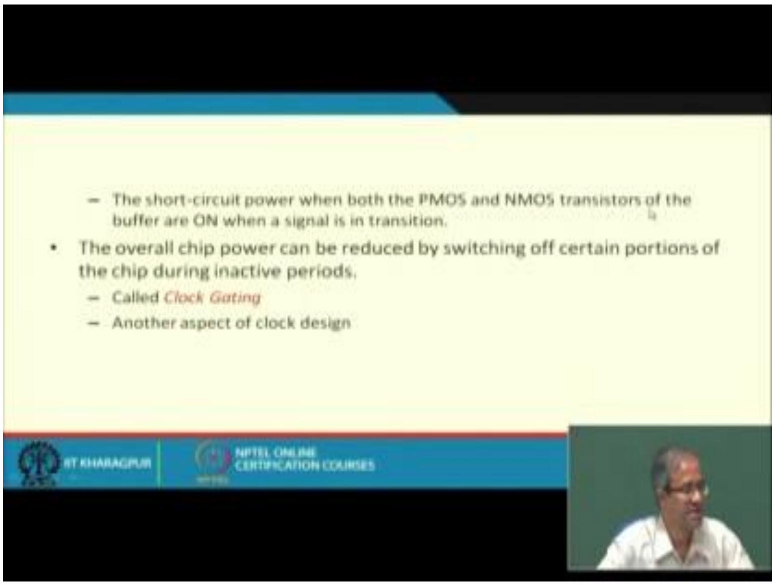


Clock Power Dissipation

- The clock signal switches twice every cycle.
  - It consumes a major portion of the power for a high-performance processor (approx. 40%).
  - A low-power clock tree implies a low-power chip.
- Power consumption has two major components:
  - The dynamic component: $C_L V_{dd}^2 f$, where C_L includes the capacitive load of the entire clock network including
    - The wires of the distribution
    - The clock buffers
    - The flip-flop loads at the leaf nodes

Regarding clock power dissipation I have just now mentioned. The clock signal switches continuously twice every cycle one from low to high and high to low, and it can consume up to 40 percent of the power of the chip. So, this is quite a major issue. So, if you can implement a clock tree that consumes a low power, this can also imply that a chip will be a low power chip. So, the power consumption will be having two major components one is of course, the dynamic component, depending on the frequency of switching f the to the power will be given by the capacitive load, square of the power supply and the frequency. So, the wires of the distribution, the clock buffers they will determine C l the flip flop loads they will all determine the capacitive load. So, if you can reduce the capacitive load in some way, your power consumption can decrease.

(Refer Slide Time: 27:05)

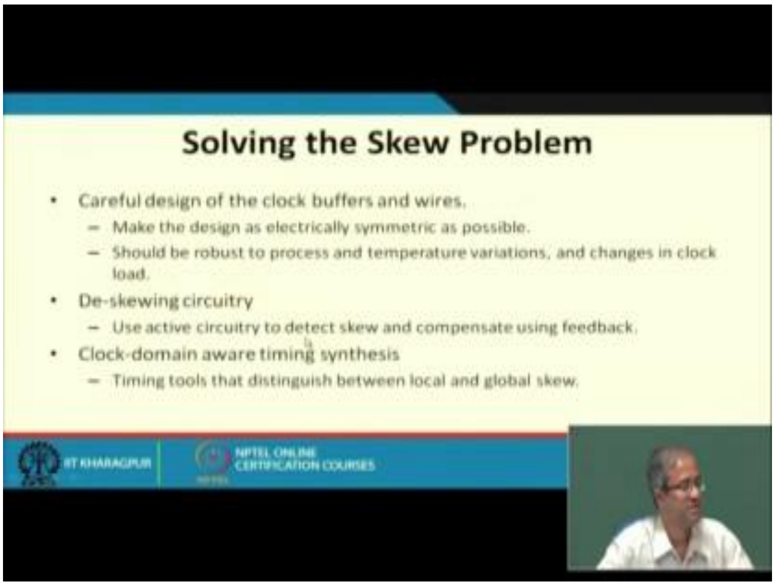


- The short-circuit power when both the PMOS and NMOS transistors of the buffer are ON when a signal is in transition.

- The overall chip power can be reduced by switching off certain portions of the chip during inactive periods.
  - Called *Clock Gating*
  - Another aspect of clock design

Then of course, short circuit power when during switching, for very short durations both the PMOS and NMOS networks in a cmos gate are on and there is a short circuit power this is called short circuit power and of course, overall chip power can be reduced by switching off some portions of the chip when it is not required, this is called clock gating well again we shall be talking about this much later this is another aspect of clock designing.

(Refer Slide Time: 27:36)



Solving the Skew Problem

- Careful design of the clock buffers and wires.
  - Make the design as electrically symmetric as possible.
  - Should be robust to process and temperature variations, and changes in clock load.
- De-skewing circuitry
  - Use active circuitry to detect skew and compensate using feedback.
- Clock-domain aware timing synthesis
  - Timing tools that distinguish between local and global skew.

Solving the skew problem there are a few rules of thumbs you can follow; careful design of the clock buffers and wires we shall look at some of these later, make the design electrically symmetric as I had said and should not be too much dependent on temperature and process variations, and also change in load it should be that way.

You can use some special circuitry, which can detect skew and automatically do some compensation some feedback circuitry based on face lock loop or something like that. So, you can have an expressive circuitry for that, and you can also have a delay aware synthesis or timing aware synthesis, where it can explicitly look at the skew problem and do some kind of clock tree synthesis that controls the limit skew to within a tolerable limit.

So, with this we come to the end of this lecture.

Thank you.