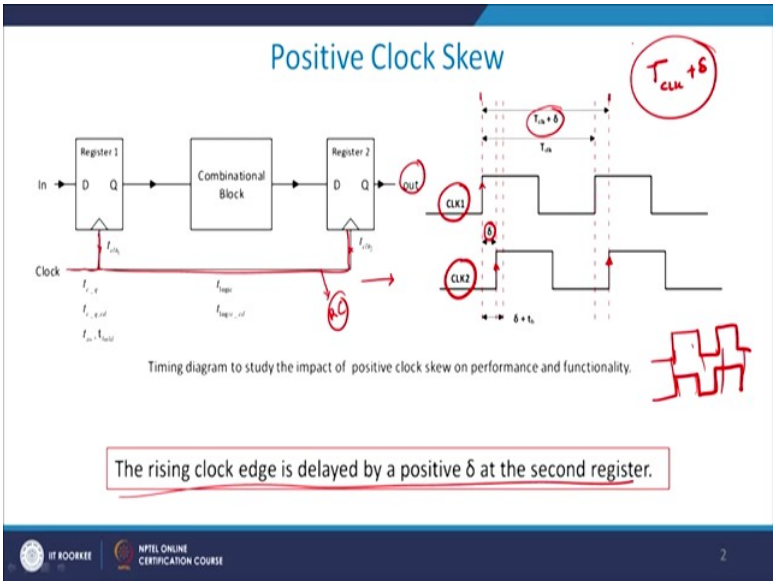
Microelectronics: Devices to Circuits
Professor, Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee
Lecture 59
Clocking Strategies for Sequential Design-II

Hello everybody and welcome once again to the next edition of NPTEL online course on Microelectronics Devices to Circuits. We start off with the second module of the clocking strategies for sequential logic and we will see what we have learned earlier and then we will go forward from the place where you have left especially the Jitter. We learn that the concept of Skew and we said that this Skew happens because the interconnect lengths at this particular point and at this particular point are different.

Which means that if a clock is routed through this much path and the clock is routed through a larger part than this will give you some RC delay and as a result this clock will be slightly delayed as compared with the first clock. So if your first clock rising edge is somewhere here and your second clock rising edge is somewhere here then we define this to be as my clock Skew, right?

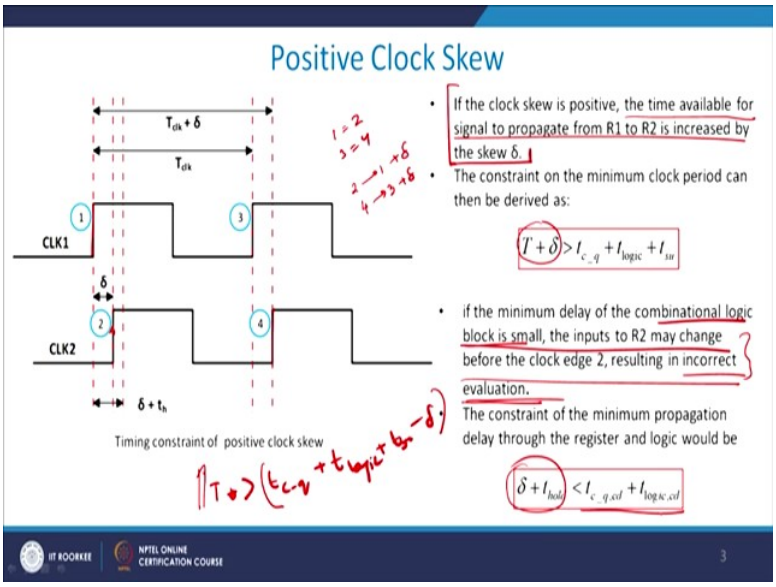
(Refer Slide Time: 1:20)



And therefore what happens is that, so this clock1 and this is your clock2 and then the distance between this point and this point is of course t_{clk} plus δ, why? Because this whole thing has shifted, so you see after clock2 only your output will be in the output state and therefore what has happened is the rising edge, clock edge is delayed by a positive δ.

So what is happening is that, you are actually evaluating at this particular point during the rising edge of the clock 2. Which means that the total time spent is T_{clk} plus δ which means that you initially had only T_{clk} to do if they were not Skew because if they were not Skewed than clock 1 and clock 2 will exactly fall each over them exactly same and therefore there will be no difference between the Skew between the 2. This is Skew you get an extra δ coming into picture and that reduces the performance of the system.

(Refer Slide Time: 2:21)



Now if the clock Skew is positive, if the clock Skew is positive which I have shown already then the time available for the signal to propagate from R_1 to R_2 is increased by the Skew factor of δ.

As you can understand since I am evaluating at edge and number 1 and evaluating at edge number 4 if there would not have been any Skew 1 will be equals to 2 and 3 will be equals to 4. In place of Skew 2 is shifted from one by a factor of δ. Similarly 4 is shifted from 3 by a factor of δ and therefore the total time is basically T. So that the reason when we write T plus δ, so if you remember our previous discussion that T should be greater than t_{cq} plus t_{logic} plus t_{setup}.

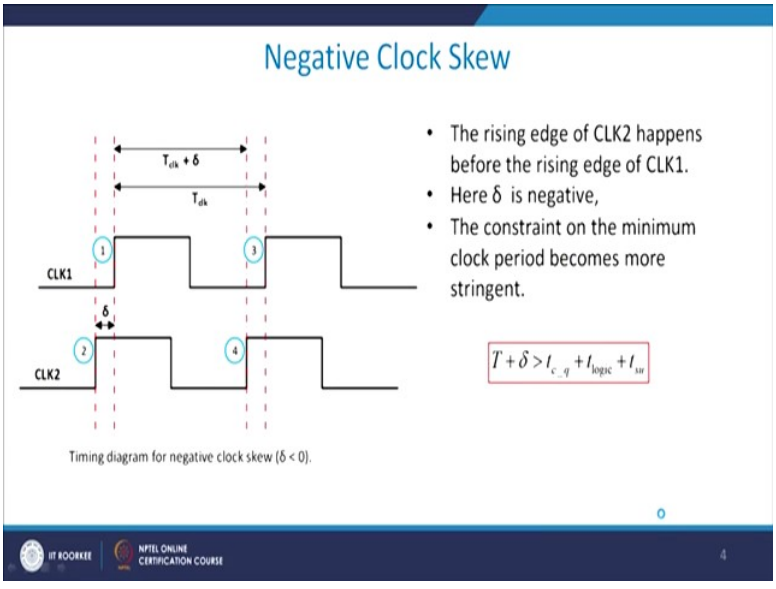
But now I say that T plus δ should be this much, so if you take δ on the opposite side. So T should be greater than t_{cq} t_{logic} and plus t_{setup} minus δ, so if the δ goes higher and higher this Skew increases then this reduces and as a result your performance becomes better and better. In the sense that you are now able to achieve a larger, faster clock.

So the frequency improves but the cost you pay for it is that that there might be a problem, that is what I am coming to the second point. If the minimum delay of the combinational logical block is small the inputs to R_2 may change before the clock edge 2 resulting in incorrect evaluation. So though you have gained in terms of higher clocking speeds in terms of because T has increased but the price you are paying for it is basically that let us suppose your combination logical block.

Which was there between the 2 registers, if the delay is very-very small, so what has happened is, that if the register are 2, if the minimum delay of the combinational logical block is small the input to R_2, right may change even before the rising edge of the clock this second, are you getting my point? So your output was synchronize with respect to clock 2. Now what has happened is, since the delay of the CLB is very small it has actually come before the rising edge and you are not able to evaluate it and therefore resulting in incorrect evaluation and that is the constraints which you get.

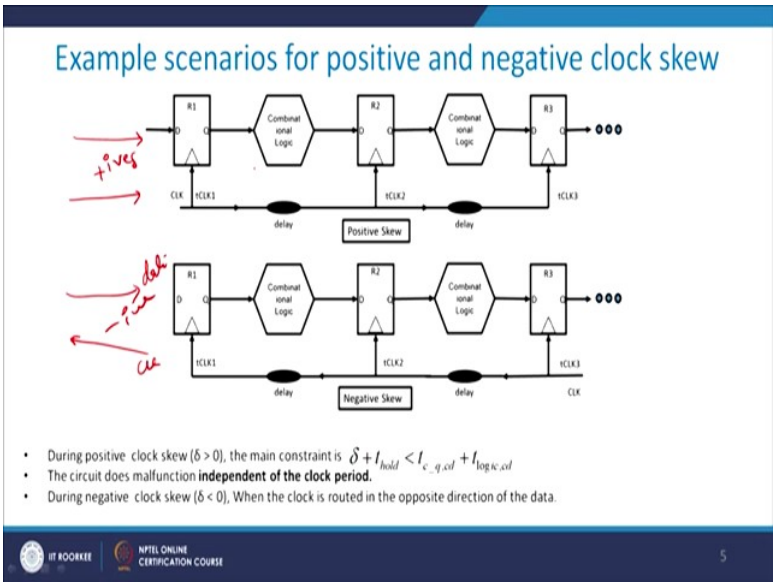
The constraints of the minimum propagation delay through the register is δ plus t_{hold} should be less than $t_{cq,\ cd}$ plus $t_{logic,\ cd}$. If it is not then there will be a problem of evaluation also for a positive edge triggered.

(Refer Slide Time: 5:05)



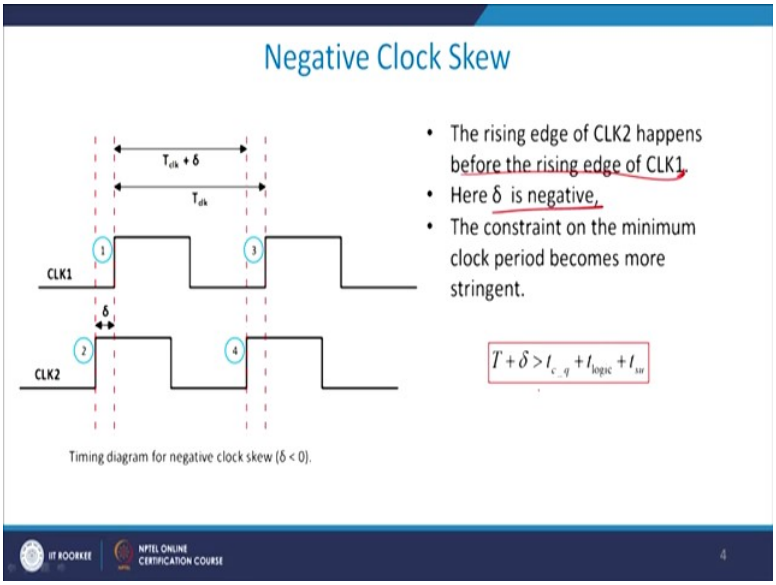
How do you define negative edge triggered? Negative edge triggered is when clock2 is moving ahead of clock1. So this is clock2 and clock1, this will happen when?

(Refer Slide Time: 5:14)



As I discussed in the previous turn that when the data, this is the data path movement and this is my clock movement, I will always get positive Skew. If my data path is opposite to my clock, this is my data and this is my clock and then we get a negative Skew, so you need to point out this particular point.

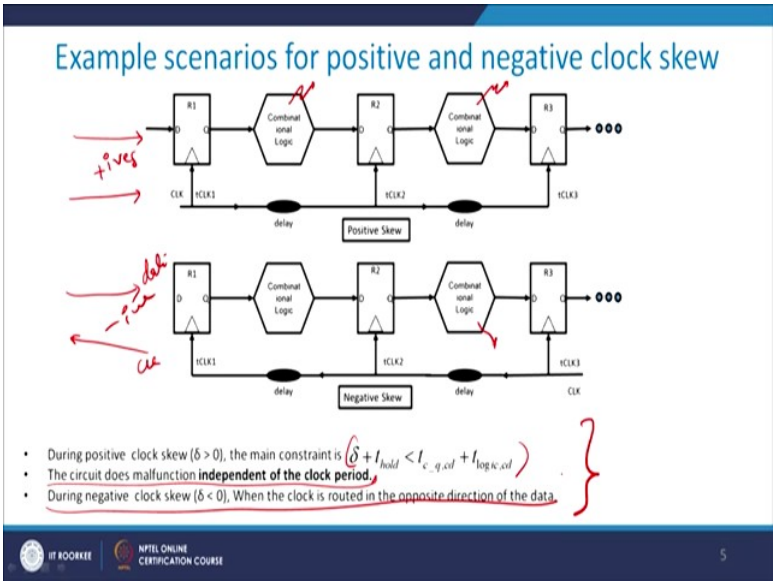
(Refer Slide Time: 5:33)



And therefore the rising edge of the clock 2 happens before the rising edge of clock 1 and therefore the δ is negative or we define that the clock Skew is negative but the constraint is therefore again as I discussed with you T plus δ should be greater than or equal to this. So the positive or the negative Skew does not influence by overall performance. It only tries to make

the point that the CLB should have a minimum, the combination logical block should have a minimum delay so that the edge 2 is properly being refer to.

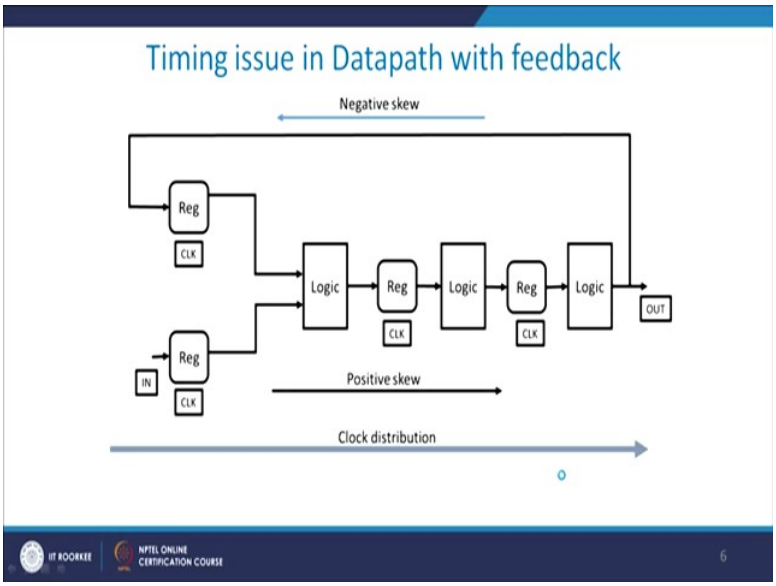
(Refer Slide Time: 6:06)



This is what I was talking about in my previous turn also that δ plus t_{hold} should be less than equal to $t_{cq,\ cd}$ plus $t_{logic,\ cd}$. Now therefore the logic does malfunction independent to the clock period, you got the point. If the combination logical block for example here this case, right? Or this case, right? Or even this case, if this combination logical block relatively have a very very low delay then the clock has already arrived but your data is not available to be evaluated, so get a wrong data output in the case.

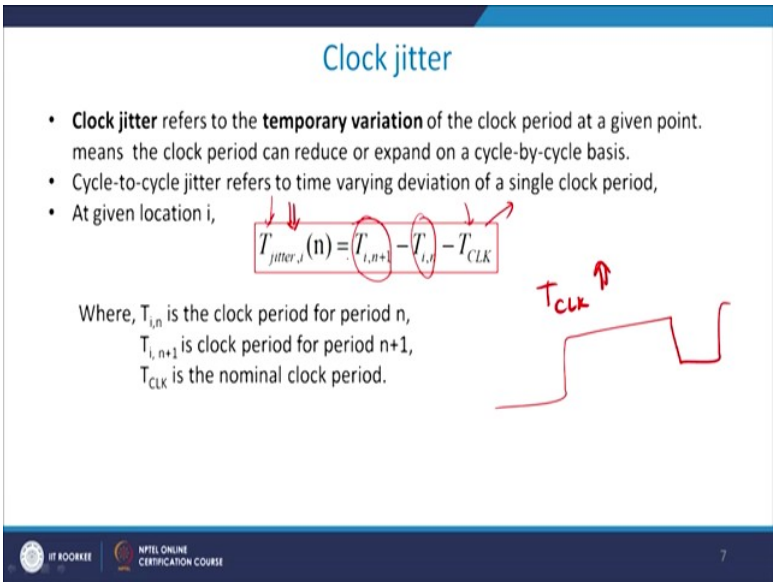
So as I have discussed with you in the negative Skew the clock is always routed in the opposite direction to the data. Whereas in the positive Skew they are always routed through the same data.

(Refer Slide Time: 6:51)



Now clock distribution if you look very carefully, as I have discussed with you. As you move from input to output if you move in the direction of the data flow you will always start getting positive Skew and if you move in the opposite direction to dataflow you will get a negative Skew, right? But both will help you in terms of performance and reduce delay at the cost of wrong evaluation. So functionality might be under question if this is there.

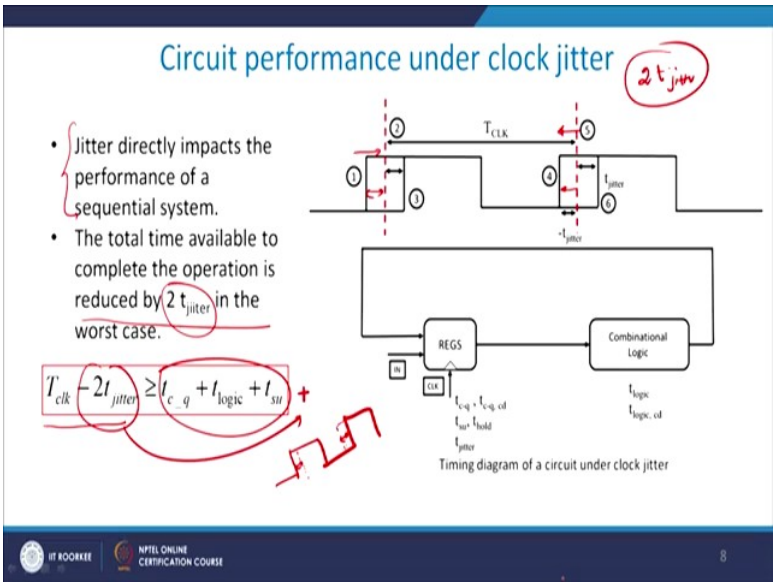
(Refer Slide Time 7:16)



We also discussed clock Jitter that is a temporary variation and we define that to be as $T_{i, n+1}$ minus $T_{i, n}$ minus T_{clk} and so on and so forth that is defined to be as T_{Jitter}, so if the clock period is very-very high, large then I will get a lower value of T_{Jitter}. So if my T_{clk} is large which means that I have a large T_{clk} I will get a lower Jitter but then you can understand large T means a lower frequency.

So a lower frequency will allot you lower Jitter, so a lower Jitter can only achieve through a lower frequency operation of the system.

(Refer Slide Time: 7:59)

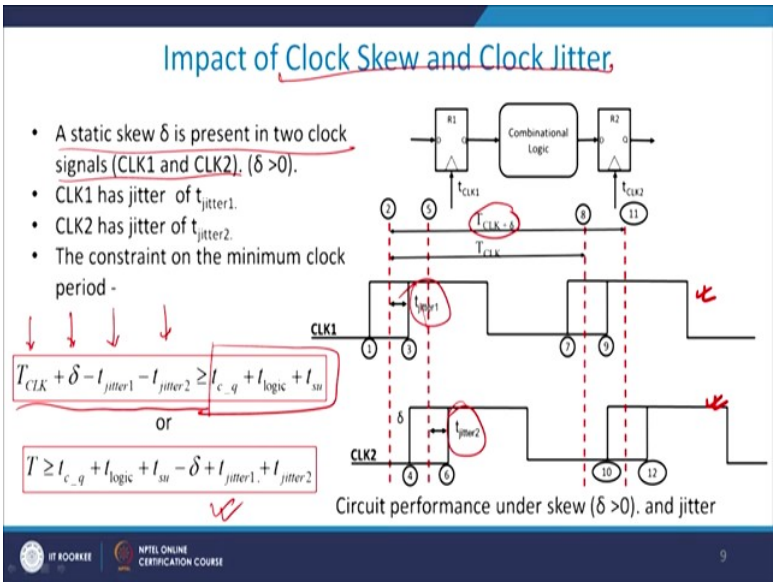


So as I was discussing with you the Jitter directly impact the performance of a sequential circuit. The total time available to complete the operation is reduced by 2 Jitter in the worst case. As you can understand see what will happen is, if by far this 1 moves to let us say 2 and this 5 moves to 4, let us suppose. So this moves by $1t_{Jitter}$ and this moves by $1t_{Jitter}$, so the maximum movement one can handle is $2t_{Jitter}$.

And that is what is written that the worst-case Jitter is basically $2t_{Jitter}$. Why $2t_{Jitter}$? Because if this is your clock, right? And this is your rising edge and falling edge of the clock then let us suppose this is moved by this direction and this is moved by this direction temporarily. So I lost 1 this much and I lost this much, so this is $2_{tJitter}$ which you get. So I get T_{clk}, T_{clock} minus $2t_{Jitter}$ should be always greater than this value.

Which means that if you take $2t_{Jitter}$ on the right-hand side it is always adding up. So if you take this to the right-hand side it will always add up which means that Jitter will always lower the frequency of the output signal, right? Whereas Skew in fact improves it to an extent but then at the cost of functionality of the chip in question. Whereas Skew will actually increase the overall delay means lower the frequency actually.

(Refer Slide Time: 9:32)

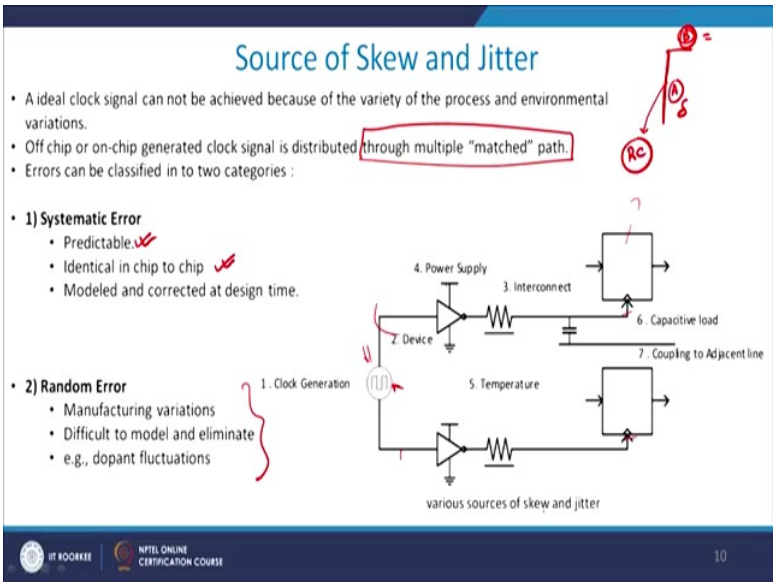


Now if I take both clock Skew and clock Jitter together, let us see how it impacts the overall circuit. As I discussed with you a static Skew δ is present between 2 clock signals clock1 and clock2, so let us suppose this is clock1 and this is your clock2, this is T_{clk} for the first case and this is the T_{clk} for the second case, right? And then you have T_{clk} plus δ coming into picture because of the clock Skew which is there with us.

And there is also a Jitter which $t_{jitter1}$ here, right? And you have $t_{jitter\ 2}$ here, so there are 2 different clocks and 2 different Jitters. So the constraint on the minimum clock period is the T_{clk} plus δ minus t Jitter 1 minus t Jitter 2 because the worst-case scenario was twice to t Jitter. But now let us suppose that the Jitter of first clock and second clock are unequal then we write T_{clk} plus δ minus $t_{jitter\ 1}$ minus $t_{jitter\ 2}$ must be greater than equal to this value.

Or I can shift everything on this side and we take minus δ plus $t_{jitter\ 1}$ plus $t_{jitter\ 2,}$ so this is a generic form of having Skew and delay. Skew will always try to improve the frequency, Jitter will always try to reduce the frequency in a sense.

(Refer Slide Time: 10:50)



Now let us come to this source of Skew and Jitter and we will see how it works in that sense. An ideal clock signal cannot be achieved as I discussed with you because let us suppose you have a problem, the problem is something like this. If the clocks are generated generally in the off chip because on chip the power dissipations are very high, it is getting heated up and therefore you generate clock off chip.

Once you generate clock off chip and then you route the signal from outside the clock to inside the chip and driving an active device or maybe a combination logical block or a register then you will see that the routing distances is not equal. So if am moving from this point to this point and I am moving from this point of this point, suppose this is path A and path B.

Path A will obviously have a larger RC delay as compared to path B and therefore I would expect to see a larger Skew here as compared to point B that is what is written and therefore

though through paths if they are matched paths which means the RC δ are equal then I will not get the Skew. If they are unmatched you will always get a Skew. There are 2 types of error therefore.

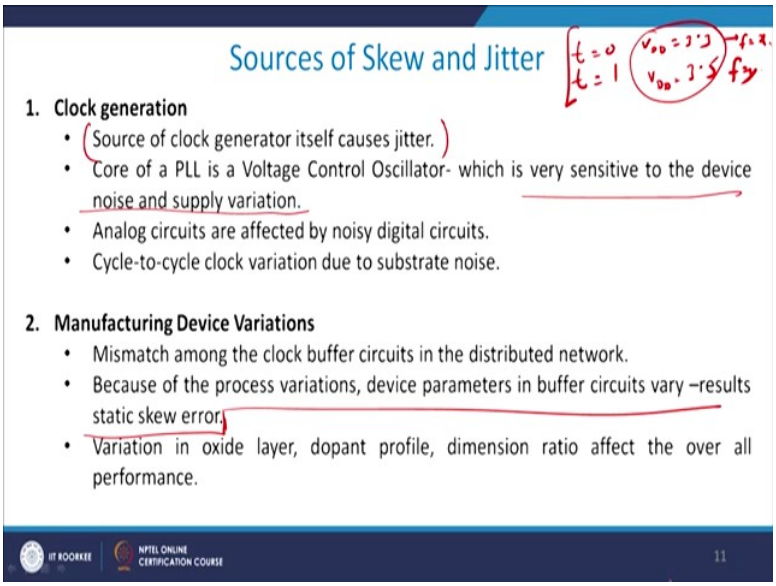
One is known as the predictable error which you can predict by virtue of the mathematical treatment in terms of how many, what is the interconnected length and so on and so forth. And then there identical from chip to chip, so if you take the same length in any of the IC which you are fabricating the delay will almost be the same in this case. And therefore it can be modelled and corrected at the design time itself.

So at the design time you know what is the length of the interconnect and therefore you can predict what the value of your Skew will be and I can therefore give leveraged to my t_{clk}, so that the Skew is taken into consideration. However, what is a random error? When you have manufacturing defects or manufacturing variations and they are difficult to model because these are random in nature for example dopant fluctuation is one of them.

Now as I discussed with you therefore if you look at this, I am generating an example of various Skews and Jitter. So if am generating a clock here and then I have a power supply through which and I have an inverter let us suppose. And then there is an interconnect with drives register here then what will happen is, even if these 2 lines are exactly equal, right? But the clock loads might vary which means for example that a single clock might load 6 devices together.

And as a result the clock loads will vary and therefore the capacity of loading will vary and therefore Skew will vary and that there is reason you will have various Skews and Jitters in reality.

(Refer Slide Time: 13:43)



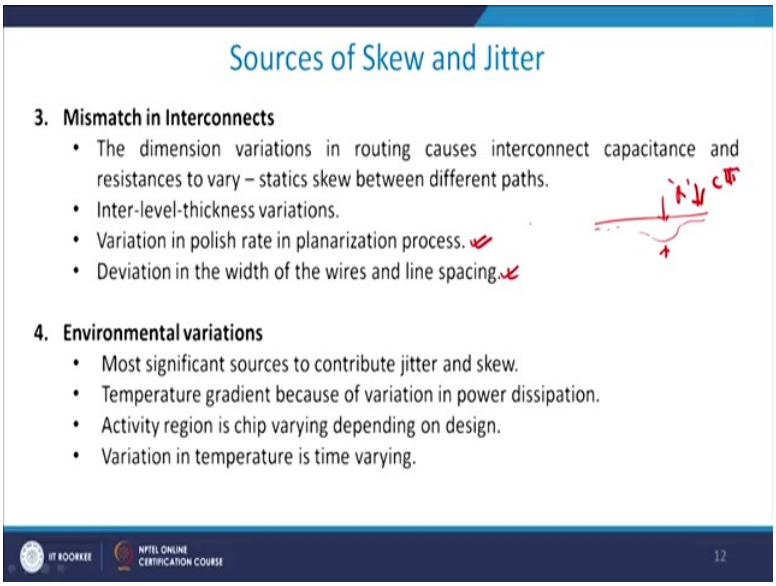
Now the source of clock generator itself causes the Jitter for example generally we use VCO which is voltage control oscillator or even PLL which is responsible for generating the clock but the problem is these are very very sensitive devices and they are prone to noise and supply variation. So even if your V_{DD} which you have supplied is changing by even a fraction of amount that frequency will change slightly and therefore that gives you a Jitter.

So suppose you are at t equals to 0 your V_{DD} was equals to 3 volts or 3.3 volts suddenly at t equals to 1 this V_{DD} goes to 3.5 volts then this will give a frequency correspondence to x. Suppose this goes to y which means that in time domain or temporal domain you have shifted from X to Y because of just because of the variation in the Valley of V_{DD} and that is what is written in the second point, bulleted point here.

Similarly cycle to cycle clock variation due to substrate noise, so you will have substrate noise you will have thermal noise especially in analog circuits and they will give rise to change in the frequency. As I discussed with you random variables or random functionalities your manufacturing device variations because of process variation device parameters in buffer circuits vary and results in Skew error.

For example if your oxide thickness varies, your dopant concentration varies, your W by L ratio varies in all these cases in Manufacturing defects you will have a variational threshold voltage for example which results in output frequency variation and therefore they will be changed in the variation of the value of a Skew and Jitter.

(Refer Slide Time: 15:21)



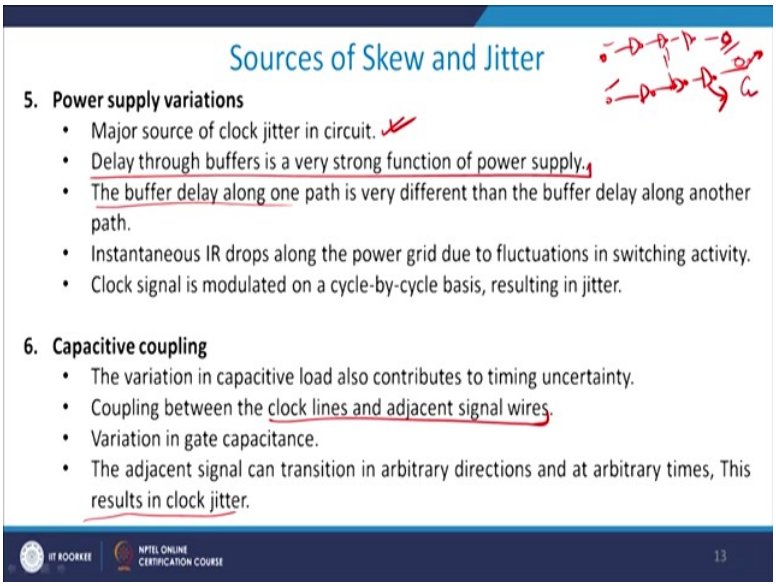
Similarly the third is basically the mismatch in interconnects this is the most prominent one and the reason is that the paths even if they are matched there might be some variation when you actually fabricate it on chip and that might result in the mismatch in this Skew and Jitter in reality. Similarly one example which people have been facing is the variation in planarization process.

The deviation, the widths of the wires and line spacing so though you think your line is like this internally the line maybe something like this. So here the width is large and therefore the resistance value here falls down and your capacitance value rises up. So you never know what is happening across the interconnect here in reality and as a result your Skew will be or Jitter will be there in that case.

Environmental variations are the most important contributors to the Jitters and Skew variations for example if there is a temperature gradient or if there is a variation in the power you automatically get a higher Jitter because at larger temperature variation implies at that particular point you will have large changes in the resistances of the wire which you are using as a result your Jitter will change and that is an important part which people play.

The fifth part is basically as I discussed with you earlier, power supply rejection of power supply variation. This is a major source of clock Jitter in circuits.

(Refer Slide Time: 16:51)

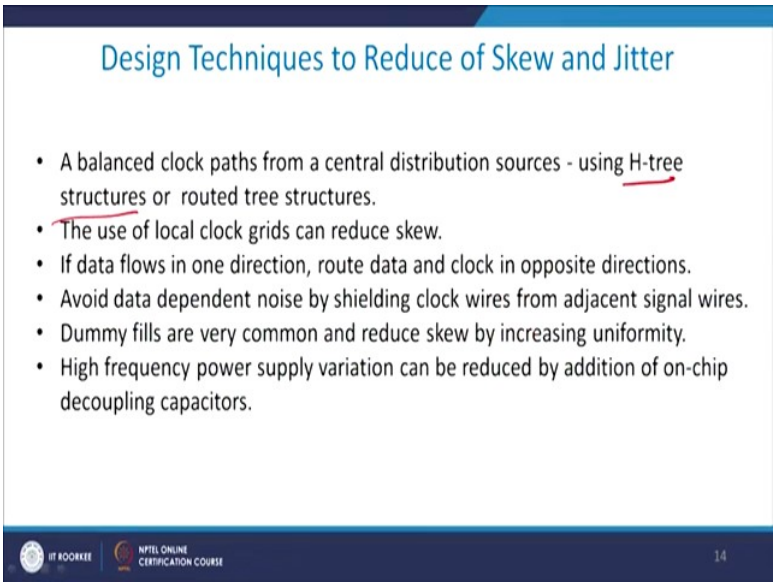


Similarly the delay through buffer is a strong function of power supply we have studied this when we were discussing FO4. Similarly the buffer delay between any 2 path, so I would delay here and I am putting buffers here.

Even if the buffer lengths are equal between 2 paths we will see that, even if the buffer lengths are equal it might be true that this path has got a higher delay as compared to this one because here it has to drive an external load which is of high value as compared to a load here. So your CL value or logical effort is typically very high in this case as compared to this case and the delay therefore is larger.

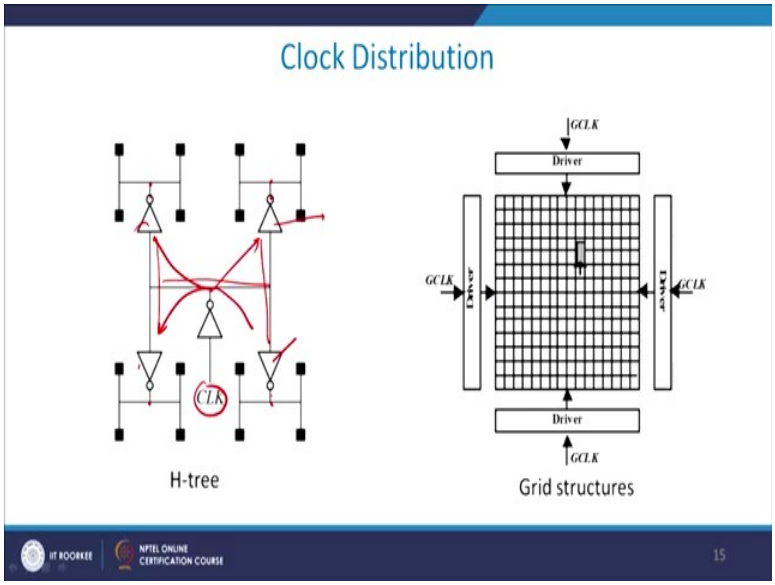
Capacitive coupling. Capacitive coupling basically means that you have 2 adjacent lines carrying the data, right? Then there will be coupling between the clock lines and the adjacent signal wires and this will vary the overall gate capacitances and this results in clock Jitter in a sense. Instantaneous loading, capacitive loading goes on changing because you do have a talking between the 2 wires very close to each other, so you have a problem of electromagnetic interference in the sense.

(Refer Slide Time: 18:03)



Okay how to reduce the Skew and Jitter. The best option available to you is try to make the clock as symmetric as possible and one of the example is basically H-tree.

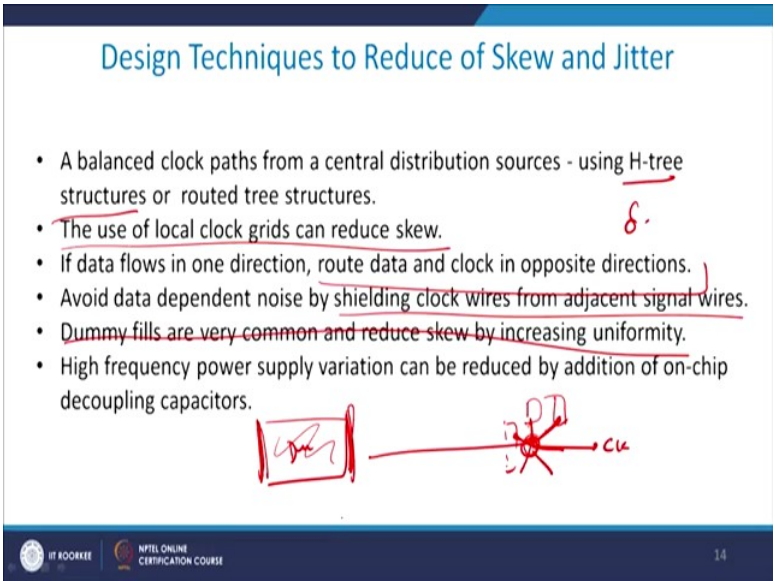
(Refer Slide Time: 18:16)



H-tree, I may have a diagram, yes. This is the H-tree which you see, so I have generated a clock here and using an inverter I generate the clock and then this is H. so this is your H and therefore this distance, from this point to this point, this point to this point, this to this and this to this are exactly same. Assuming that the delay of these inverters are exactly the same.

The clock will reach at these points exactly the same instant of time and this is known as a clock distribution for H tree clock distribution.

(Refer Slide Time: 18:58)



Now another methodology which people have adopted is that rather than routing clock from external source down to the whole register modules, what we can do also is that, let me route a clock from here to here and then locally if you have large number of blocks here, here, here then you route the clocks from here, so you make a hub and a spoke. So this hub will be the basic clock and from this clock you will have…

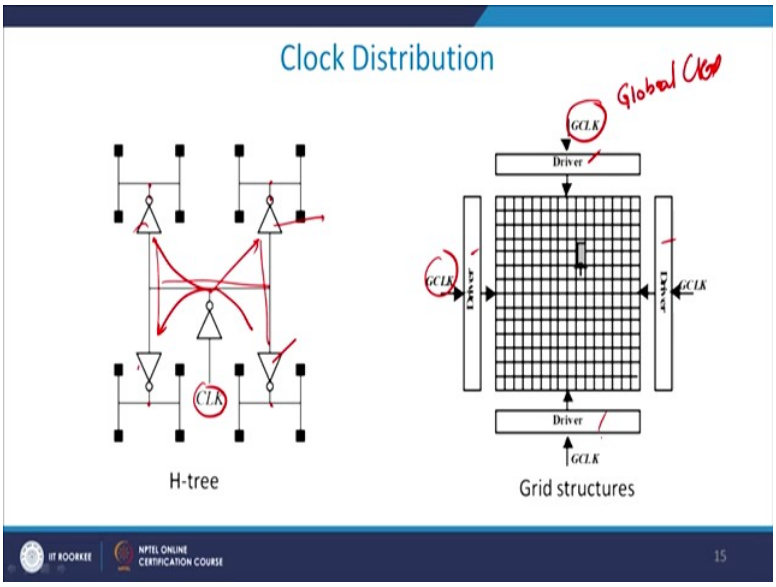
Now since this distance were very-very low or very-very small they are effect on Skew will be relatively small as compared to having 6 different paths from external clock to this path it is always advisable that you should have a single path terminating near the blocks and from there you route smaller wires to these clocks in that way you can almost eliminate 80 to 90 percent of this Skew.

As I discussed with you if the data flows in one direction route the data and clock in opposite direction you automatically get a negative Skew. Once you get a negative Skew you are an advantage in the sense. Similarly you can shield the adjacent wires as I discussed with you, you can shield the clock wires from the adjacent signal wires this will result in noninterference and therefore you will automatically get better reduced Jitter.

One methodology which people adopted is at layout level was Dummy fills. Dummy fills are primarily, the areas where you have blank spaces, you have a silicon what you do is, that you fill there with some dummy structures, so that between 2 edges there is less amount of interference. So let us suppose I have got one structure here, another structure here this is empty space I can have easily interaction.

What I do? I feel this up with a dummy fill in that act as a very good logic breaker between the 2. You can add decoupling capacitors we need not worry about it and try to make your V_{DD} as stable as possible in order to reduce the Jitter problem, so I discussed with you this is the clock distribution of an H tree.

(Refer Slide Time: 21:15)

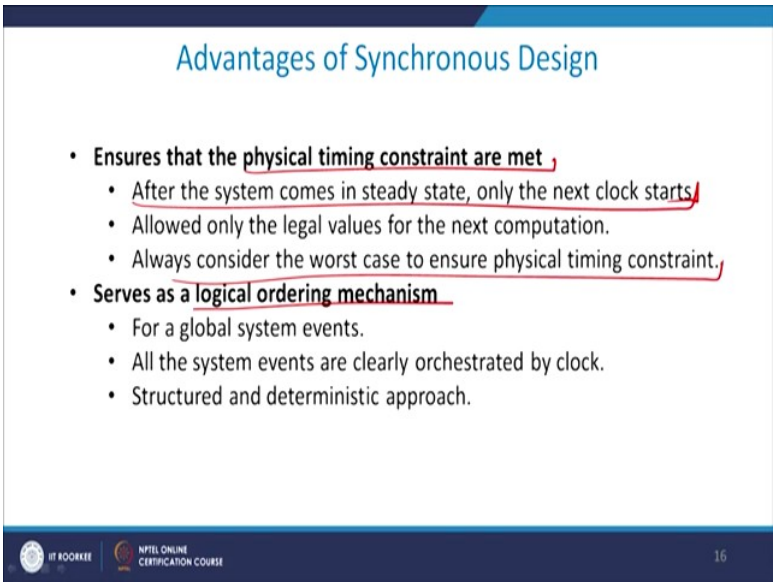


So what I do is, so GCLK is basically my global clock and this global clock actually drives the drivers here which drives the local clock. And this local clock distance is from each of the blocks are so small that you effectively do not get any delay as such.

So let me come to the last part of our talk and that is what is the advantage of synchronous design. This ensures that the physical timing constraints are met. Of course this is very very important. And second part serves as a logical ordering mechanism which means that whenever a data is being processed, then the data should be processed in a serial fashion.

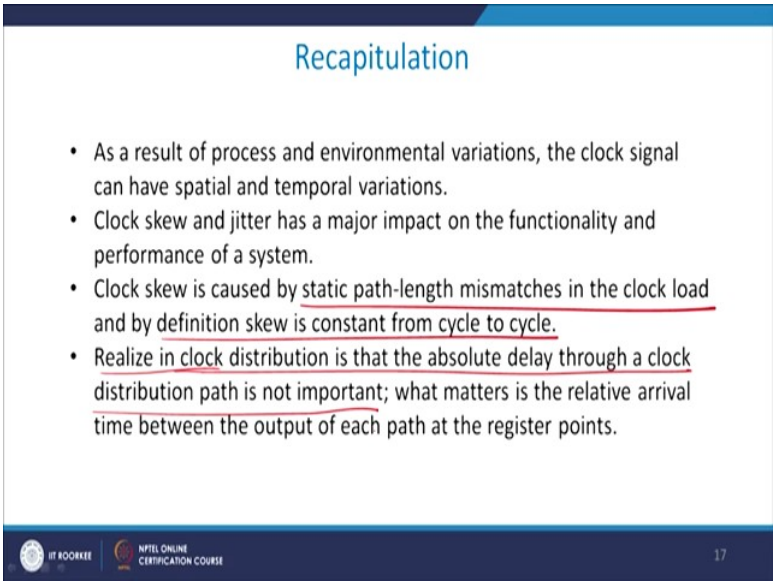
Because the first data gets processed and then second grade of data, third grade of data that is can be only done if you do a synchronous design because the clocks will be responsible for feeding the data to the register and the combinational logical blocks.

(Refer Slide Time: 22:12)



And the first one ensures to me that whenever you have clock you can store information till a point when the next clock arrives, so after the system comes in steady-state only the next clock starts and that is pretty important. Third thing is that you always consider the worst-case to ensure timing constraints. So you add t_{cq}, you add t_{logic} you add t_{setup} and the total delay your global clock should be at least larger than that. The time period for that and therefore the frequency should be less than that in any case.

(Refer Slide Time: 22:47)



So let me recapitulate the whole lecture which we did for the 2 parts of clocking sequences. We saw that what is the clock Skew, we also saw what is the clock Jitter, they effect the

functionality and performance of a system. Clock Skew as I discussed with you is caused by the static path-length mismatched in the clock load and by definition clock Skew is constant from cycle to cycle.

Whereas in case of Jitter it is not it varies from cycle to cycle. Clock Skew is positive if the data and the clock move in the same direction whereas it is negative if the data and clock moves in the opposite direction. Realize in the clock distribution is that the absolute delay through a clock distribution path is not important. It is more important that what is the relative separation in time domain between 2 clocks edges energizing my output and that is very very important in each case.

So this is what we have done and that takes care of the whole module as far as sequential logical block is concerned. I thank you for your patient hearing. Thank you.