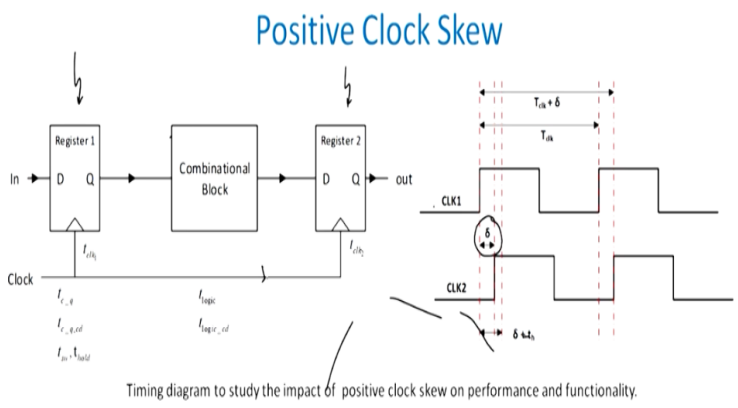
CMOS Digital VLSI Design
Prof. Sudeb Dasgupta
Department of Electronics & Communication Engineering
Indian Institute of Technology Roorkee

Module No #07
Lecture No #35
Clocking Strategies For Sequential Design – III

Hello everybody and welcome to the NPTEL online certification course on CMOS digital VLSI design we will start with the next module of the clocking strategies for sequential logic in the previous we have seen what do you mean by clock skew and How it is related? We will carry forward in this discussion in this module and see apart from the skew what are other problem which you face right?

(Refer Slide Time: 00:52)



Timing diagram to study the impact of positive clock skew on performance and functionality.

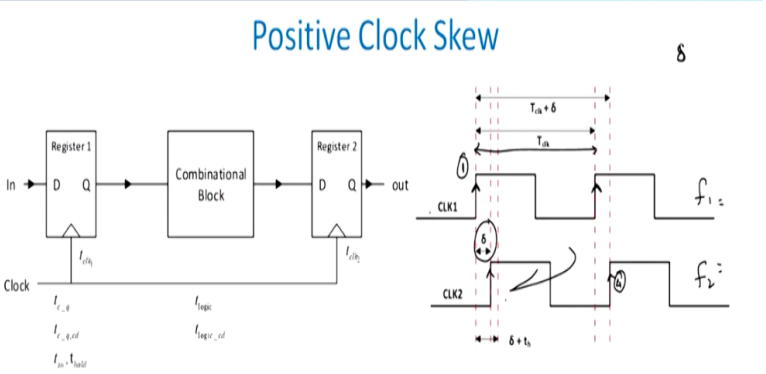
So as I discuss in previous turn that for the positive so what happens is that if the clock is delayed by a factor delta which you see in the front of you. Which is this one if the clock is delayed by the factor of delta we referred this as a positive clock skew right we have a so which respect at the second register please understand the rising edge is delayed by a positive delta at what at the second register so the first register have actually evaluated the data right.

But the second register is yet to evaluate it and therefore the clock actually shifted in the external extra interconnect path so on and hence so forth so you have a positive logic there or positive delay in the clock so if the direction of the clock right the movement of the clock and the

physical placement of a registers are the same direction then for later register you get higher skew so let us suppose you have another combinational block here and you have a register number three here later on.

Then register number three will have a large skew as compared to register number two right. So this is the positive clock skew which you will entrain we can also have a negative clock skew wherein if the direction of the data path and the clock are just reverse in direction you will have a negative skew right so let see how it works out?

(Refer Slide Time: 02:10)



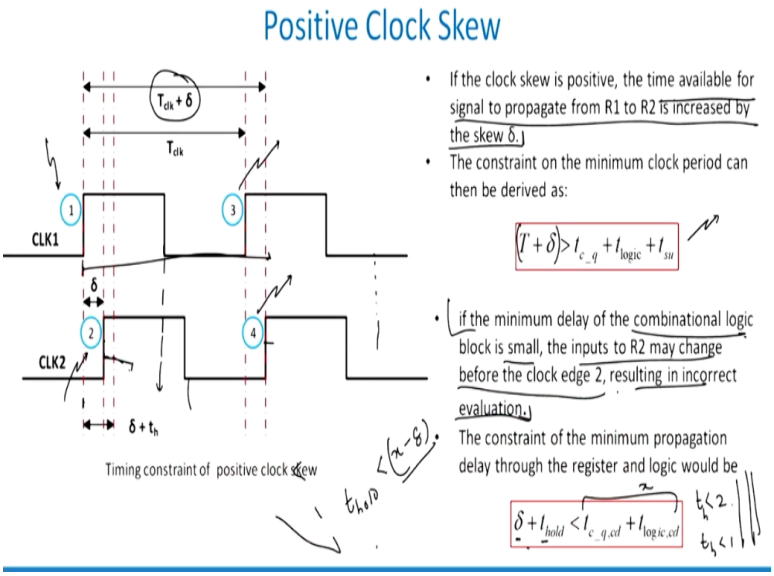
Timing diagram to study the impact of positive clock skew on performance and functionality.

The rising clock edge is delayed by a positive δ at the second register.

So if this delta right as I discussed with you and this is my tck so this is my t clock period which you see so for rising edge of first clock to rising edge of the second itself similarly this is your clock period. So you see the clock period still the same for clock one and clock two they are extract the same the clock frequency is perfectly match F1 is extract = F2 what is not matching is possible the phase because the rising edge is shifted to right.

Now so if you look very carefully you have actually sample to data at say edge number one right and have to sample the data when at edge number 4 next register 2 as a sample edge number 4 so if you look very carefully higher the value of delta more time you are giving to the system in order to make it work fine even if you increase the delta right this is no problem as such we will see to that later on right.

(Refer Slide Time: 03:06)

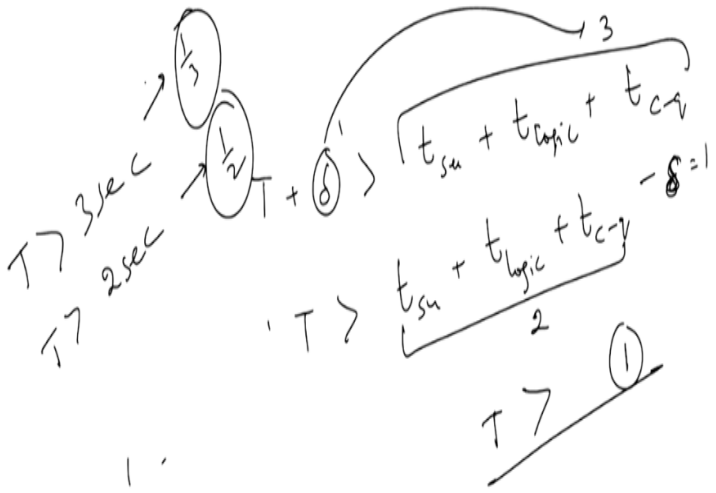


Let see therefore no next slide how its works out? Let me therefore write down that if the clock skew is positive that is what I was talking about? This is edge number 1 which is seeing edge number 2 is the second clock rising edge. Edge number 3 is the first clock rising edge and edge number 4 is second clock rising edge write this are four edges which you see now if the clock skew is positive right.

Then the time available for the signal to propagate from R1 to R2 is increased by the skew equals to delta I am clear like you get the point that since you are increasing the value of delta so the time available for you to propagate from this edge various sampling the data till this edge, till this edge so basically t clock + delta is total time available with you so if you remember the previous discussion.

Where I was discussing with you the t should be greater than tcq + t logic + t setup there is small change in the inequality and the new value is that t + delta should now be equal to greater than tcq minus + t clock + t setup.

(Refer Slide Time: 04:32)



So this is the new value with you see now you see if this is true that t + delta should be greater than t setup + t logic +tcq then t should new inequality you should be t setup + t logic right +t2q – delta because this delta transfers to this side you get - delta right which means that in fact if you look very carefully a positive skew age your sequential logic because if it is – delta it basically means let us suppose this was 2 second now this will be just 1 second.

Suppose delta = 1 then 2-1 will be 1 therefore timing constraints are relaxed right so what did you told that let us suppose this whole things was 3 second and this one second so as per the initial discussing t should be greater than 3 seconds right. Now with skew t should be greater than 2 seconds right that is the whole logic which I am trying to do which means that I am allowing for so this will result in a frequency of 1 / 3 this will result in a frequency of 1 /2 right. So obviously this frequency is large as compared to this.

And therefore having a skew actually increases the frequency of operation for the device right but looks very simple but you cannot go on increasing the value of delta to your liking as much as you like there will be other problem in functionality there let me a problem of race conditional so no and hence so forth right you cannot do that because if you go on increasing delta infinitely what might happen is if you go on increasing delta basically a shifting edge number 2 away from edge number 1.

Let us suppose edge number 2 reaches this edge right then what will happen edge number 4 will reach where edge number will edge 4 will reach somewhere here right H4 will reach somewhere here where the problem will be that therefore that H4 which is this particular point the data available to the sample. So functionality works only for few values of delta you cannot go on increasing the delta as you go on doing it right.

So if the minimum delay of the combinational logical block is small inputs to R2 may change before the clock edge to resulting incorrect valuation is suppose is clear that if the minimum delay of the combinational what combinational logic block this combinational logical block this one. If this combinational logical block delay or block is small the minimum delay is small enough what might happen is that.

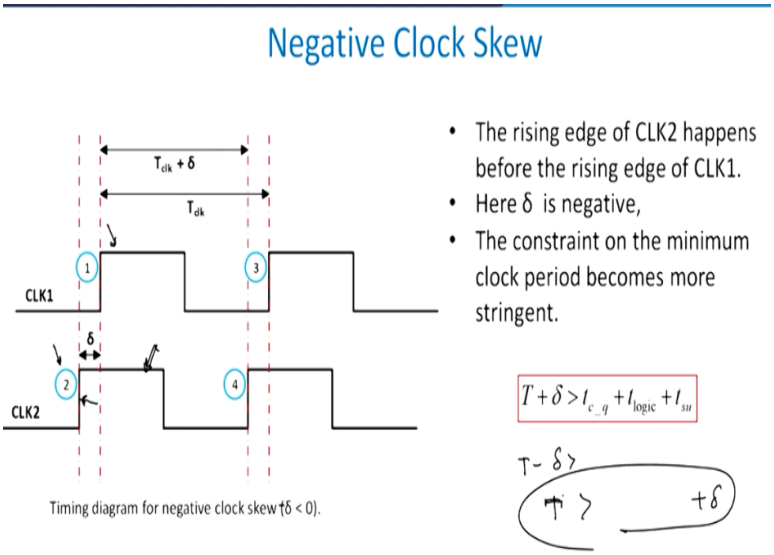
Since it is very small so even before the rising edge of the clock of R2 your data is actually changing at the input of your clock 2 right and that will give you a incorrect evaluation right similarly if delta is available which means the skew is there what I get from there is T delta + t hold should be less than Tcqcd + T logic cd why is it true why we are adding delta to T hold the simple reason is the clock is just laterally shifted in time domain delta whatever the application you are holding for t hold will just added over the delta right so therefore I will get t hold is equals to less than whatever this small tt.

Let us suppose this is x this is x-delta though your clock period is getting much better but your t hold is getting worse and worse. So let us suppose your tcqcd + tl logic cd was initially 2 seconds so t hold was actually less than t seconds now when delta equals one this will actually t hold will be less than 1 second which means that the constant on t hold is now much more stringent as compared to the previous case right and that is the reason the t hold will be much smaller in this case.

So therefore that is that I was saying you cannot increasing the value of delta as much as you want because though you are having a better capital T which is the clock period of the pulse of the input clock but you lose in terms of the t hold constraints and therefore there might be condition when the data will be wrongly valuated here even with larger value of data. So this is basic idea of the basic concept behind the positive clock skew and we have already explained to

you what the positive clocks skew looks like and why is it like that in real sense right we have already done that to a large. Now let me come to the next negative clock skew is when the clock 2 which is the second one this one.

(Refer Slide Time: 09:05)



The rising edge of that happens to be earlier than the rising edge of clock 1 right so this is clock 1 and this is clock 2 and the rising edge of the clock 2 happens to be earlier as compared to clock 1 and there is therefore it is basically shifter to the left in time domain so it is coming earlier in that case delta is basically negative right. So you always your reference is basically clock 1 right and therefore anything before that it is clock 2 right when the data is negative this will be replaced by t – delta.

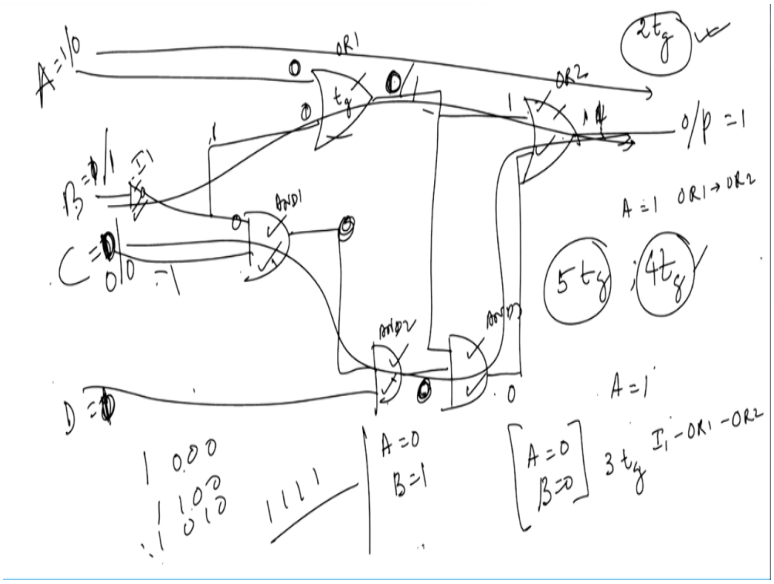
So when becomes t so + delta here on the right hand which effectively means that your time period as to be larger than a particular value. So in a negative skew condition you actually end up having a words value of capital T available with you that is the major sort of a problem area for for negative clock skew was compared to positive clock skew. So as I discussed with you positive clocks skew will always be good to you provide the value of delta does not go on increasing indefinitely because that will restrain your t hold.

Similarly when negative clocks skew always have a problem with capital T because the restrain in capital T will be more stringent negative skew which is – delta. If you look back in the previous slide so if you look back at the okay leave the previous then you will see that this we

have been assuming or you have been entering the fact that generally the clock skew is positive you will have an issue when the timing constrains can be relaxed but keep in mind that the combinational logical block which you see here will be actually combinational logical block will actually have various issues associated with it.

For example and it can give you some false path I will explain to you now what the false path combinational logic is and how did you find a critical path. Let me show you a false path so we are talking about the sequential path where the combinational block and we will show to you that certain inputs gets redundant right and therefore in that path finding out the delta as got no meaning I will just show you that basic concepts.

(Refer Slide Time: 11:19)



So let me draw for you the path basic simple circuit right and I will show to you how it works out so let me say this goes to this right and then you have a another AND gate here the output of the AND gate is fed here and you have a D input here and the AND gate what are the input is C right and then you have a inverter static inverter here which goes this this point and then this goes to this and this goes to so I have go A B C D I have got OR1 let us suppose OR2.

Let us suppose AND 1 AND 2, AND 3 and suppose inverter 1 right so this is the basic logic diagram which you see in front of you right let us suppose with A = 1 let us suppose what will happen and everything else is available to you at this is my primary output so this so I have AND

gate and this AND gate is sort of fet I am sorry so this OR2 will be fed here just forget about this part and so you have AND gate which fit into another.

So this is my output and these are my 4 input primary available to me and this is combinational logical block which you see now when A = 1 right independent of BCD output be always 1 so what is suppose I assume that all the gates have equal delay including inverter so this delay will be suppose this is each as I define it to be as t gate tg then the total delay of the critical path which will be defined as this will be nothing but 2tg because of you look if A = 1 independent of what the values of BCDR this will be giving yo u 1 here this will be giving you 1 here and then OR gate will give 1 here output will be one irrespective of the values BDCD fine.

So you see if the data would have been 1 0 0 or 1 1100 or 1 0101 or anything 1111 anyone of this combination is 1 then you have problem that output will only be available to you after two critical path it is for other it is redundant . If I take A = 0 right and B =0 with A = 0 if this is 0 here and B – 0 also I get here and therefore this is one if this is 1 this will also be equals to this will go to 1 here and therefore output will be 1.

So under the condition that A = 0 and B = 0 the critical path available to you is basically this so how many you will have 3 tg fine is it clear when A =1 and other are whatever the value the total critical path delay is hastily goes 2 tg is the delay of individual block when you have got A = 0 and B = 0 then the critical path is B to OR1 so the critical in this basically I1, OR1, OR2. In the first case it was just OR1 and OR2 and then output.

Now let us suppose I have a path A = 0 and B = 1 if A=0 B=1 implies that A=0 with this will be 0 B will be equals to B=1 which means that this will be 0. So this will be 0 this will be 0 means then what will happen is then it will depend up on the values of C and D. So let us suppose C=1 then I will get output so if this is B=0 C=1 I will get 0 here Then if D is also equals to 1 then I will get 0 here if it is 0 this will be 0.

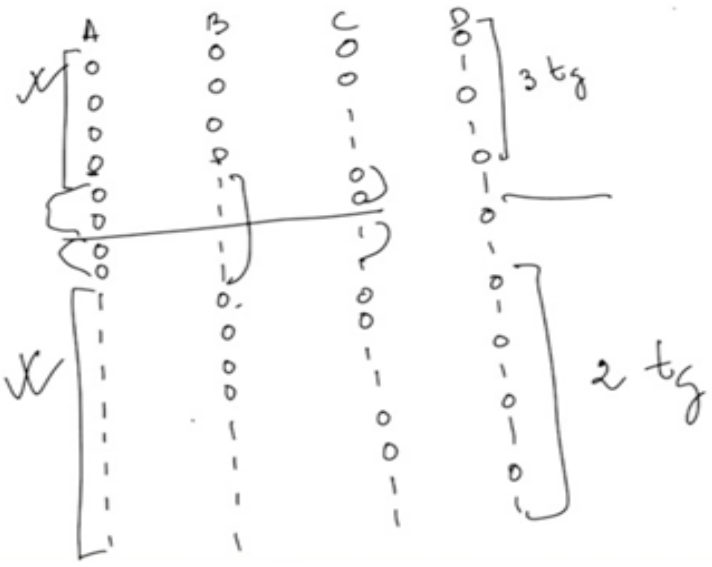
This is 0 this is in this case, If B=0 I will get 1 here and if it is 1, this will be 1 and therefore 1 and 0 will give you 1, output will be 1. But then, what is my critical path now? My critical path now is this, through this, and output will equals to 1. Similarly, if D if C=0 and D=0 right. If both

are 0 if all the 4 quantities are 0, then quite interesting thing happens that output will be only depending up on this and this.

Whereas, when C=1 the output will depend up on this which means a it will depend up on AND1, AND2, AND3 and OR2. When D=1, it will also depend up on AND2, AND3 and OR2. So what is the critical maximum critical path available to you is basically if 1 tg, 2 tg, 3 tg, 4 tg, 5 tg. So 5 tg happens when you have this circuitry into consideration, right. Otherwise, when C=1 you only have basically 4 tg available to you. So 4 tg why because 1 tg, 2 tg, 3 tg and 4 tg will be the critical path available to you.

What I wanted to tell you from all these understanding or all these basic problem areas is that I just wanted to tell you or show you that output of any combinational logical block will not only depend up on the logic structure. But it will also depend up on the logical inputs which you are giving, right. So the type of input which you are giving is in this case. And you can do a small home works of now if you want to and take up this circuit for all 16 all 16 combinations, you please try to find out the values which is available to the values you.

(Refer Slide Time: 17:27)



If you If you do that it is quite interesting that if you can do something like this I A, right, B, C and D right. So I have got if the 16 combinations, right. So I will get 0,0,0 right and these are there 8 and then 1, 2, 3, 4, 5, 6, 7, 8 be then will have 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 and then C will have 0,0,1,1, right and then you will have 0,1,0,1,0,1. Now I have a discussing just now that

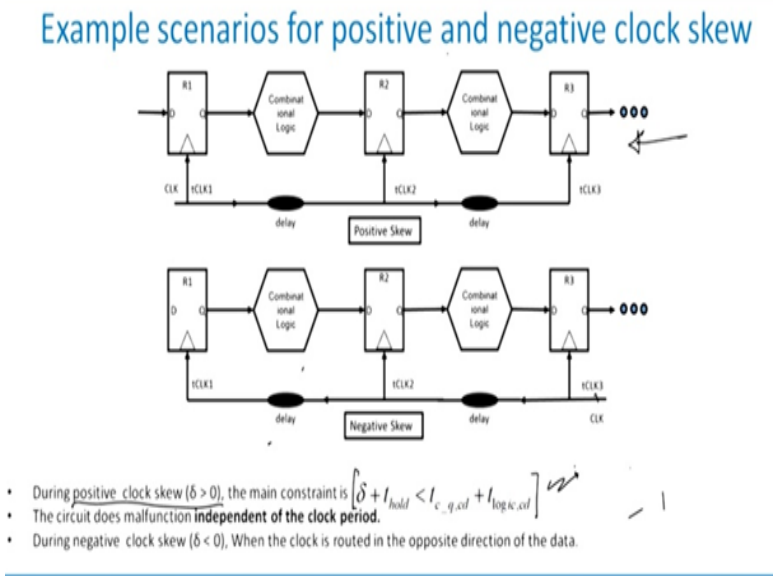
under the condition when A=1 this till this part, this will give you 2 tg as your output irrespective of the value of B, C or D right and then it will get propagated.

With A=0 and B=1, right or A=0 B is also equals to 0 which means this part, you will get how much? 3 tg right. 3 tg is the maximum value because 1 inverter and 2 OR gates you will get. With A=0, B=1 with C=0, right C=0 and so I will get this 1 block here C=0 and I will get another block here. So if you look at C=0, what will happen? At C=0, if you put 0 here, right and in that case, if you take a so just a minute if you take, for example C=0 and B=1, right, B=1, C=0 if you take in that into consideration. Then what I get from there is with B 1, B 1 and C=0 let us suppose.

With B 1and C=0, B 1 means this will be 0, C=1 means this will be 0. Is this is 0 D is also equals to 0 will give you 0 here and then all 0 will get propagated here. But the problem is that since B=1, I will get a 0 here and therefore I (()) (19:38) always equals to 0 here, right. So you see very carefully that that for this set and this set it is already fixed. For this set depending on the of C either 0 or 1, you will have 4 tg in 1 case you will have 5 tg in other case. So that is how you will do.

So depending upon time of logic you are using a n combinational logical block. You do have a problem of this positive clock skew. So this is the positive clock skew basic idea we are also discuss the negative clock skew. We are discuss positive clock skew, we are discuss the negative clock skew. So timing diagram for the negative clock skew if you see is minus delta is less than 0.

(Refer Slide Time: 20:28)



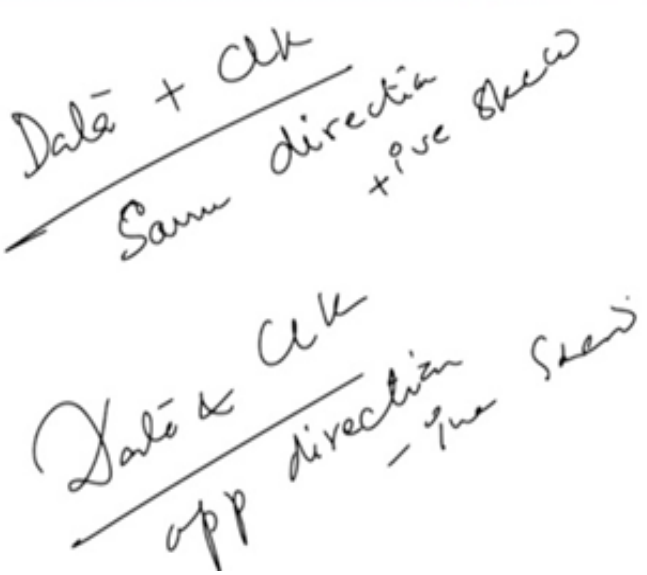
And we now discuss about examples scenarios for negative and positive clock skew. So you see what I was discussing just now to you. That is is clock 1, clock 2, clock 3, right. And the data or the clock is moving from clock 1 to clock 3. Then we differ these 2 is the positive clock skew, right. So clock 2 will be slightly more positively skewed as compared to clock 1. And clock 3 will be more positively clock skewed as compared to clock number 2.

So that is a major difference between a positive. In negative skew otherwise, you see the I am moving clock 1, clock 2, clock 3 from left to right of the screen. But I data is delivered from right to left, right. So what will happen is that clock 3 will have automatically will come early as compared to clock 2 which will come further early as compared to clock 3 clock 1. So there will be a negative skew always available to you in all constraints.

So during the I told during the positive clock skew therefore the main constraint is this, we have already discuss this point. If you do not have this constraint right, then your circuit will always malfunction, right your circuit will always malfunction independent the clock period which means that the set up time and whole time evaluations should not should be always kept in mind. That you do not do a set up time and whole time evaluations which means that in whole because if you remember when I was discussing with you the positive skew you we discuss that the delta positive delta is always good.

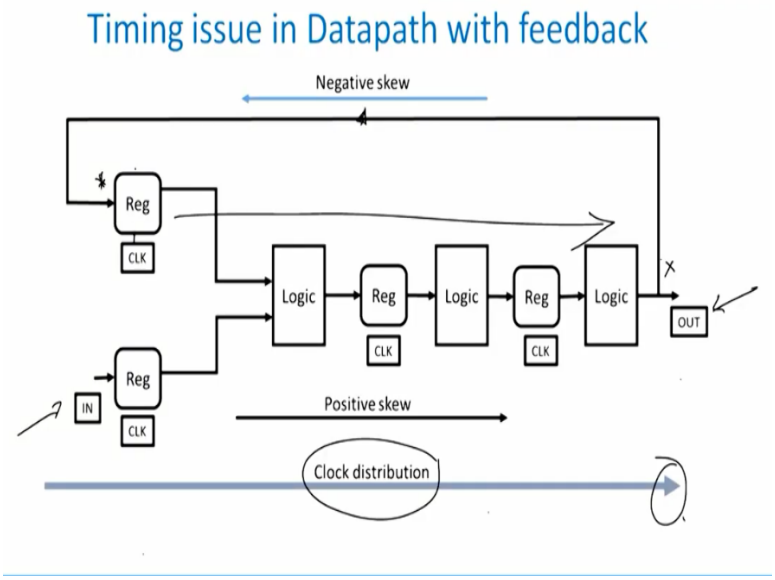
I said it is good but then beyond a particular point it is not good because then its starts to hit your t hold. As a result, you cannot work beyond a point which means that your functionality of the sequential logic will beyond the question now, right. And therefore you will get an automatically a wrong functionality in the output side, right. And that is the problem area which you see during the negative clock skew as I discussed with you the clock is routed in the opposite direction to the data. So the clock if the data is flowing from register 1 to register 3, the clock is moving from the opposite direction.

(Refer Slide Time: 22:32)



So if the data and clock, so just keep in mind if the one simple rule that if data and clock, right are moving in the same direction, then we automatically get what is known as a positive skew, right and if data and clock, right are moving in opposite direction then we always get a negative skew, fine. This is very important that when the move in the same direction, we get a positive skew and opposite direction they get a negative skew. If you look overall, then in a data path, typical data path where your feedback sequential logic of all feedback.

(Refer Slide Time: 23:17)



It is always advisable that when you do a feedback you try to do a negative skew because the data is moving in this direction, right and when you move the data is moving in this direction, right. If the clock which is movement in this the data and the clock are moving in the same direction primarily meaning that all your feedback forward arrangements in the forward direction will give you positive skew whereas all your feedbacks will give you a negative skew, fine.

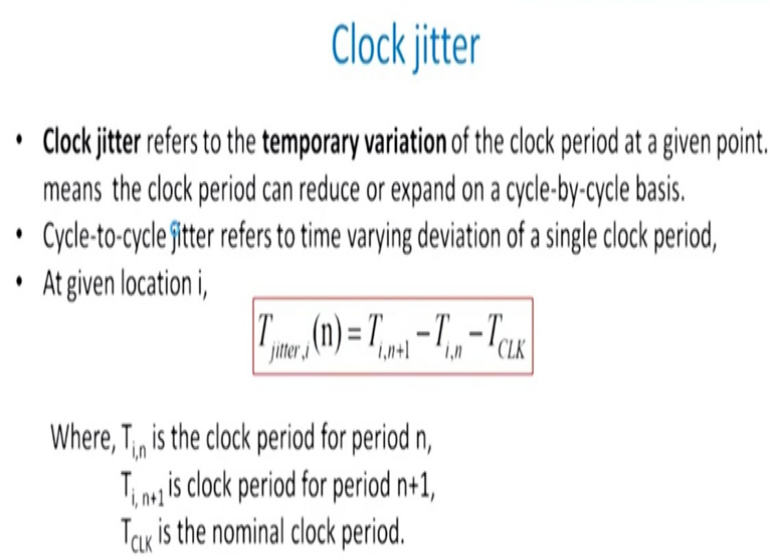
So this is quite interesting and quite important that if you look at the clock distribution path with the bold arrow we have shown here. This is the clock distribution path and it is moving from input towards the output. So this is your primary input here and this is your primary output here. It is moving from primary input to primary output from left to right, it is moving and it gives me a very good evaluation of the logic.

But since the clock movement and the logic movement data movement in the same direction, we have positive skew in in the positive direction. But when you have a negative feedback from output to input, you will have a negative skew which means that if you route your signal from this point to this point. You have to ensure that what will happen is that if you do like this and this is a clock available here, the data might be available much later even before the clock later than clock.

So clock is actually evaluated, right. But your data is coming later on, right and that's the issue a idea which we do not want to keep in (()) (24:42) keep do not want to have it in the feedback

path, right. And therefore most of the cases we have to be very careful about the feedback path in any design as far as we have a concerned right.

(Refer Slide Time: 24:54)



With this we will take up the clock jitter in the next module of our sequential logic. We understood clock skew and we understood what are the issues of clock skew what is positive clock skew, negative clock skew. We will take up the clock jitter in the next turn or in the next module, okay thank you.