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

Module No # 06
Lecture No # 28
Sequential Logic Design - IV

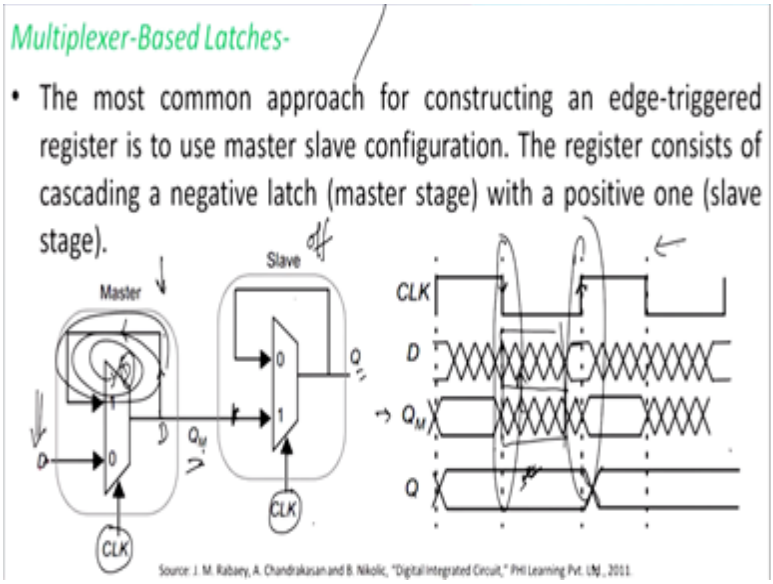
Hello everybody and welcome to the next module of NPTEL online certification course on CMOS Digital VLSI design and we will be having this module named as sequential logic design – IV. So, in our previous modules we have understood what is the sequential logic and how it is different from combinational logic. We have also seen what are the limitations of sequential logic in terms of maximum operating frequency.

Which is not there in a combinational logic we have understood the basic concepts of various constraints which is basically the setup time and the hold time constraint we have also understood the basic fundamental principles of H triggering and latch. So difference between a register and a latch we have understood we can have a positive H triggered and negative H triggered design.

We have also seen that you can have therefore sample the data in register through a your H triggering and you can sample your data during in a latch level triggering. So these are the few things we have already discussed, in today's module we will take up the further issues which are governing these the sequential logic and see how it can actually modulate it or change it for updating the maximum usage of the sequential logic design.

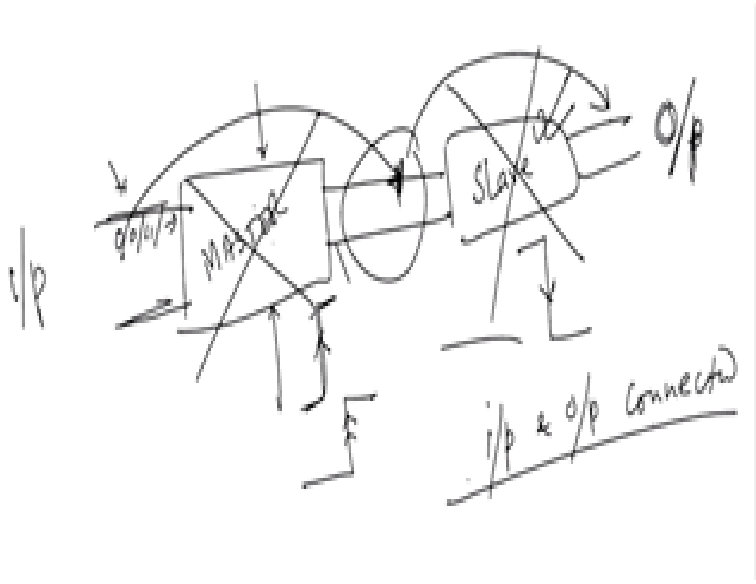
As you have already understood the sequential logic is primarily accessed in memory. And therefore if we even a basic memory will require a sequential logic understanding and its appreciation.

(Refer Slide Time: 02:19)



So to understand that we have already done that so just to give you a continuity sake, we are explaining to you what is a multiplexer based latch so this is the most common H triggered register is used. So this is the most common H triggered way of storing a data at a particular point of time and then so on and so forth so if you see very clearly this multiplexer-based latch is primarily looking like this, then you have a master right the concept is a concept is like this.

(Refer Slide Time: 02:49)



So if you have mastered, suppose I will name it as master and you have a slave which is like this just name this 2 modules which will be right and you have this is my prime input and this is my output coming out now of you see, the problem in the previous concept was that you require that the data should be stable till at least the set up time of the register which means

that sometimes before the rising edge of the clock your data has to be stable right and after the rising edge of the clock.

Also it has to be stable performing some duration of time for proper evaluation so that sets limit on the frequency right. So what we try to do here is something like this, so if I have a input right and your master is transparent to input say at some clock. So say for the example of rising edge of the clock right. But latch is not transparent for the rising edge of the clock then for the rising edge of the clock input will be sampled via the master to the output of the master.

So when the clock is rising in the rising edge of the clock the master will actually sample the data from where from the input. So you have 0, 1, 0, 1, 1 so on and so forth coming here. It is sampled each bit and it will store here at the slave. But since slave is actually not transparent at this stage so this is not working at this stage right so what will happen is output is basically latching on to the previous value, old value of the system right.

So though you are sampling your data here, since your latch is slave is not working then you play then you your placing the data at the input of your slave for all duration of time. During, let us suppose the negative half cycle when the clock is going low let us suppose the slave becomes transparent but then the master becomes opaque. So master then stops the input coming from here and whatever data was stored here will now be sampled by the slave by output side this is the basic concept of mark based latch or a multiplexer based latch.

You will ask me how, why is it done, it is done purely to negate the concept of setup time and hold time violations right. Because when your master is on and your slave is off there is no how directly input and output connected at any point of time right. So at no point of time your input and your output and output are connected right. They will be always disconnected and therefore you will always get a finite time and you can actually remove the setup time issues here right.

So just to come back to therefore to the original slide so you term this as master. So this master is basically let us suppose a negative H triggered register and you have a slave here which is basically a positive H triggered so as you can see here I have a given a clock here right and this clock same clock is given to the slave. This is my input data byte which is

coming here right and this path shows that it is a feedback loop available to me and therefore any data.

So what will happen is that during the negative edge of the clock this D will be sampled to this point but since this latch is off right. This D will be moving across this loop and therefore this loop will be storing the value of data D for infinitely long duration of time until the latch becomes the slave becomes transparent to show that, look at the timing diagram here right. So when the clock goes in a negative H triggering of the clock this is your data.

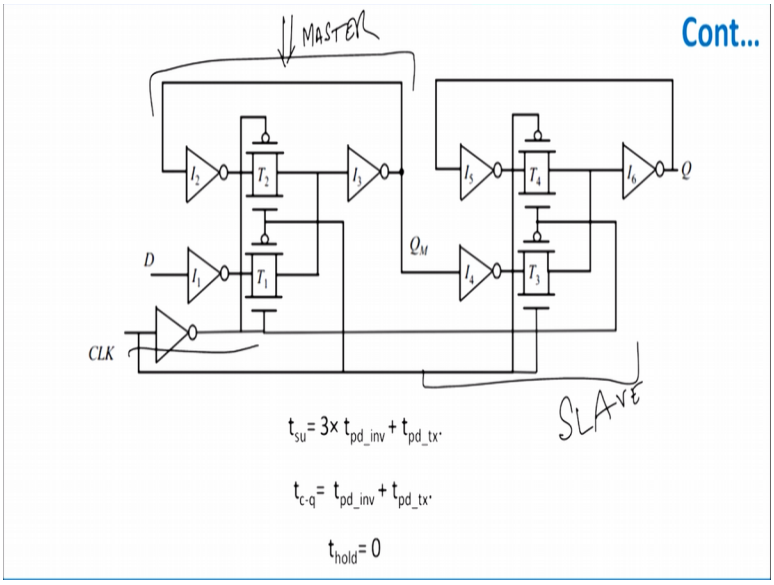
The data is getting replicated right the data is getting replicated here is it okay. At QM here and this is my, QM is the output of the master right and Q is the master, output of the slave so at that point of time Q is holding to its old data maybe 0 or 1. Whatever the data was there during the previous cycle of slave getting on it is holding the data at this particular point right.

So the negative edge of the clock, the D gets sampled to QM and this D is going through an infinite feedback loop and therefore this master is able to store the value of D at the output QM right. This positive feedback loop helps you to store the data for a longer period of time right that is the reason why we do a feedback loop. Because see a data will be basically stored in form of voltages now.

If there is some leakage or something the voltage starts to fall down you need to again refurbish the you need to rejuvenate the volt is there. So it always advisable to have feedback loop that is the reason you are having a feedback loop in the master. Now, what is happening is, now what has happened is during the rising so we are done with this part right, now we go to this part right where you have the clock rising up, rising edge of the clock.

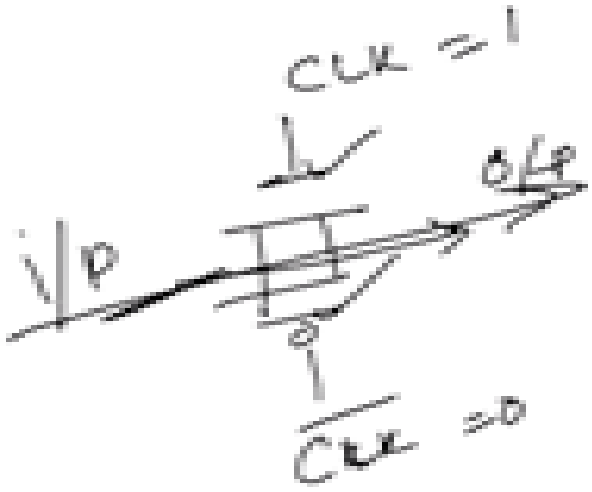
At the rising edge of the clock as I told to you, your data is stable here but then your Q shows a shift depending upon the value of your last value of your D stored here fine. So I am able to have this QM here which is therefore this is data the last value of your QM will be latched at this point and therefore when the clock goes high 0 to 1 that value will be entered into this and this Q will be stored here. So this is basically H triggered, a negative H triggered master and a positive H triggered slave configuration available to you.

(Refer Slide Time: 08:26)



And this is the diagram which is quite common and used very often in these cases. So let me show to you maybe the sorry, let me show to you the marks based design using TG using transmission gate right.

(Refer Slide Time: 08:43)



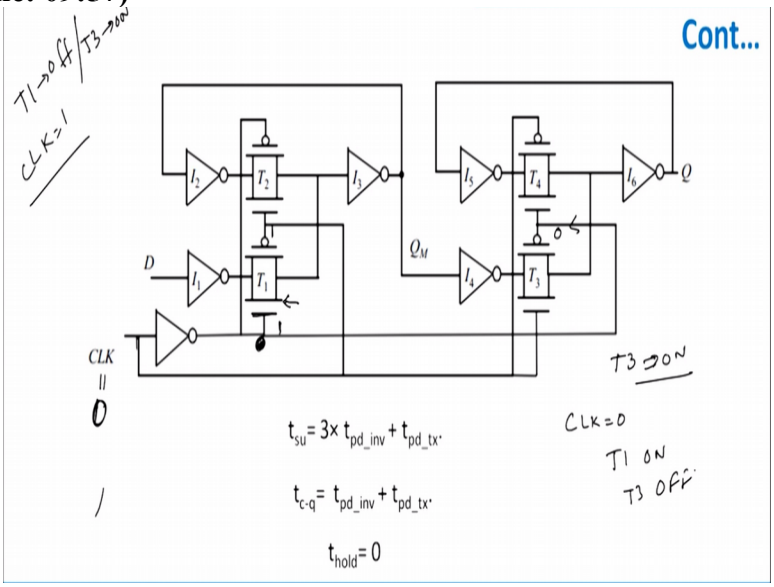
Remember transmission gate was when basically transmission gate concept if you remember was that if you have an N mass and P mass right, and if you have a P mass and a M mass and it is driven by clock and then this is driven by clock bar right. Then when clock equals to 1 and the clock bar equals to 0 it switches on and therefore both the devices are on and therefore there will be a flow and there will be a flow of current from the input.

This is the input, this is your output**.** And this is equal to your input, there will be flow of current from your input to output and the voltage will be there and the advantage here is that

you are not making the voltage fall down below a particular limit, so there is no problem the treasure voltage drop as you see a pastor system logic. NMOS pastor system logic right. So that is the basic concept.

So let us look at the basic concept here and see how it works out. So I have a TG which is T1 and T2 for the master so this is master, this is my master configuration. This is my master and this happens to be my slave this is my slave right, so I have a master and a slave available to me, this is the master which you see in front of you and this is the slave which is in front of you, now look very carefully whenever your date or whenever your clock is say let us suppose clock is high right let me just.

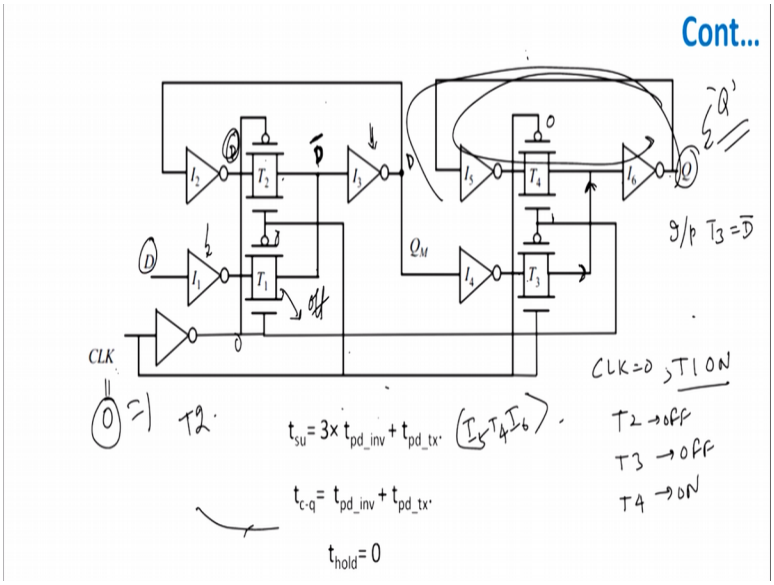
(Refer Slide Time: 09:57)



Let me put it in this manner that your clock is let us suppose is 1 right. When the clock is 1 this is 0, this goes to 1 which means this is 1 right so when this is 1 and this is 0. Since this is N mass, right therefore T1 is off right. T1 is off under the condition when clock equals to 1. I suppose you understand because when clock equals to 1 it is an inverter so I have a 0 here implies that T1 will be on and to this will be off.

But this also implies that this 1 will directly go to this thing so T 3 will be on because this 0 will be coming here right so T3 will be on right. So what will happen is that T1 will be off and T3 will be on for clock 1 right. Let us look at clock 0, if I put clock equals 0 then this is clock equals to 0 let us suppose then this will be goes to 1 right and therefore T1 will be on and T3 will be off fine.

(Refer Slide Time: 11:07)



Similarly when clock equals to 0 right, then this 0 will be transmitted here right and therefore your T1 will be on as I discussed with you. So when a clock equals to 0, T1 will be on state but T2 will be still off right, why T2 will be off because P mass is driven by a 1 state right. So when T1 is on and this is when T2 is off any data which is available here will be inverted by I1 which is a static inverter will be fed here.

And you will get here at equals to D bar right. So, what will happen to D bar, will again will through I3 it will again invert and become D here. But when it becomes D here even the so, the input of your T3 will be therefore D bar. So input of your T3 will be exactly goes to d bar at the same instance of time, if you held your clock to be equals to 0, then you are ensuring your T3 is off. So T3 off means your slave is basically off. It is not able to sample any data but.

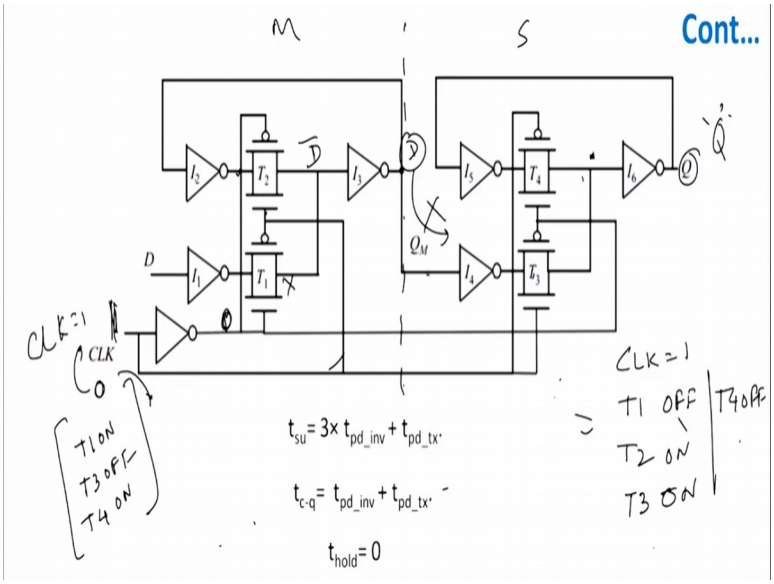
It is holding a previous data and I will show you why and how. Because if you look carefully in this case, this 0 clock 0 goes to here and therefore T4 is in on state. T4 is on, because this is 0, this is 1 right. And therefore T4 is on. T4 on implies that this feedback loop which is available to you, is always valid. So whatever your original Q value was there is been stored in the loop of I5, T4 and I6.

In this loop you are able to store the value of Q. so what will happen, if this is Q, this will go back here, you will have Q bar here this Q bar will come here again Q will be coming so there will be a closed loop available to you. Therefore you will be able to restore the value here to Q. that is the reason you will always have Q here right. But then since your T3 is off your ready, you are not getting any new data across this path right.

So your slave is effectively off, it is only meant to store or rejuvenate early data. let us look at the master at that point of time, so if you have D here which you have D bar here so when clock equals to 0. I am assuming clock equals to 0 then if clock equals to 0 then as I discussed with you T1 gets on and then therefore this is what is happening D bar gets to 0 it comes back to this place.

Here it becomes D here right it becomes D here, in the next phase of the clock when a clock goes to 1, this becomes 0 therefore T2 becomes on, when T2 becomes on D again crosses this point and reaches this point. But by that time T1 has actually become off right. T1 has actually become 1, when clock equals to 1. So let us look at very quite an interesting issue here that under the condition under that is a bit complicated to understand the initial phase but we will see how it as we move how it works out.

(Refer Slide Time: 14:19)



If you look very carefully therefore, if I want to just break it into 2 parts. I have a master condition which is I1, I2, T1, T2, I3 and you have a slave which is I4, I5, I6 and T3 and T4. So I have got 2 master and slave available to me here. This is your master and this is your slave what was the concept that if the master is on, slave will be off and if my master is off, slave will be on so vice versa will be there.

Let us look at the fact when clock equals to 0, so let us look at when the clock equals to 0, when clock equals to 0, this is 1T3, T1 is on so T1 is on right t1 is on . 0 means T3 is off right and your T4 is on right T4 is on because why because this 1 will go here and you will be

feeding here and T4 will be on and therefore T4 will be on. So T4 will be on under the case costing 0 of 01.

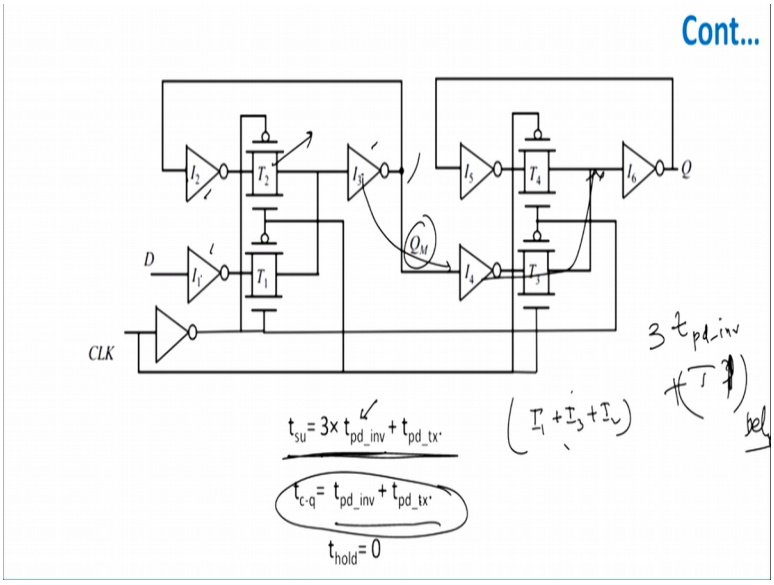
T4 equals to on and T3 equals to off means any data coming from Q1 will not be sampled by T3. And therefore T3 is in a cut off situation and as a result what will happen is whatever data that Q will be reinforced in the Q will have its original value. So this is what we have understood when clock equals to 0. When clock equals to 1 right and what happens to this place what happens is when clock equals to 0, T1 is on and T2 is off.

So when T1 is on, D goes to D bar here and D bar comes to D here but it cannot go this part because my T3 is in off state therefore it goes via this pass to I2 and is stored here right it is stored at this particular point. But understand T2 is in the off state because you are driving it by 1. Now when clock goes to again it goes to 0 here again so it goes to 0 and 1let us suppose as it goes to 0 to 1.

This clock goes to 1 and this clock goes to 0. To 0, you are ensuring that this 1 actually comes here, this 0. So 1 when your feeding 1 sorry you get 0 here, when you put 0 here then you are switching of T1. When clock equals to 1, T1 is off state right, T1 is off state but what happens to T2 is in on state and similarly what happens to T3, T3 is also in the on state.

But what happens to T4, T4 is in the off state so what has happened is I have discovered this feedback loop now any data available at this point will be accepted by T3 and it will be stored somewhere here fine. You got the whole idea that what is basically the conceptual concept of this master slave latch that master slave negative and positive H triggered design remarks based design. That you are at one point of time storing a data and removing a data and so on and so forth.

(Refer Slide Time: 17:16)



So if you look very carefully and we all we have a look into it then your this is the propagation delay for inverter, each inverter right. Why there are 3 inverters because if you look very carefully as I1, it comes here as I3 and I2 so there are 3 inverters I1, I2 and I 3 right. If you multiple with 3 times you get the total delay out of these 3 plots plus the transmission get delay by D2.

What does it mean that your set up time should be at least equal to this much. And the reason being very simple that you have to wait till the data D is being processed by all these stuffs things sorry. So this 3 so you will have what I1 + I3 + I2 delay so if assuming that all of the delay are equal so it will be 3 times the delay of individual cell right. So Pd inverter plus delay of T3. So T3 will have certain delay available to you. So T1 sorry T1 so delay of T1. Is it okay.

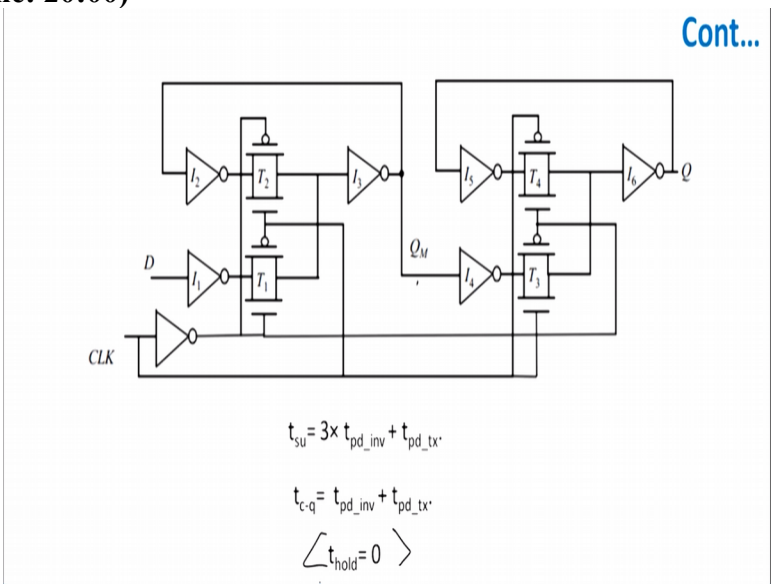
So the delay of T1 + the delay so the data will come through I1, T1 and then goes to T3, then goes to T through I2 and is stored here. Which means that you store next data which is coming at point D has to wait till this much amount of minimum time otherwise. If it comes within the same period of time it would not be able to distinguish between the old data and the new data right so that is the reason I write T set of base is equal to this one.

Whereas we write TC, C to Q delay to be equals to tpd inverter plus tpd x. Why tpd inverter because if you see very closely I just have to wait till I3 and QM for the data to actually go to this first point. So which means that when the data is actually sampled to QM you only have to wait till the delay between these two points for this data to be appreciated. And therefore C to CQ input to output basic output delay to be equals to tpd inverter plus tpd.

So, I4 + I3 delay which is available to you now we need to understand why T hold is equals to 0 right that is quite interesting why T hold is 0. T hold is basically 0, the reason being you know one of the hold time, hold time was the time which the data has to be stable after the raising edge of the clock in the deposited H triggered design.

Right, after the positive h triggered of the clock this is the minimum time you have to hold so that your data is suitably sampled in the output side and in this case it is 0 because at no point of time you are trying to make the input output transparent so there is always a logic which we will differentiate between input and output. You are getting my point. So there will be actually there will be off limit to each other, they will not be equal.

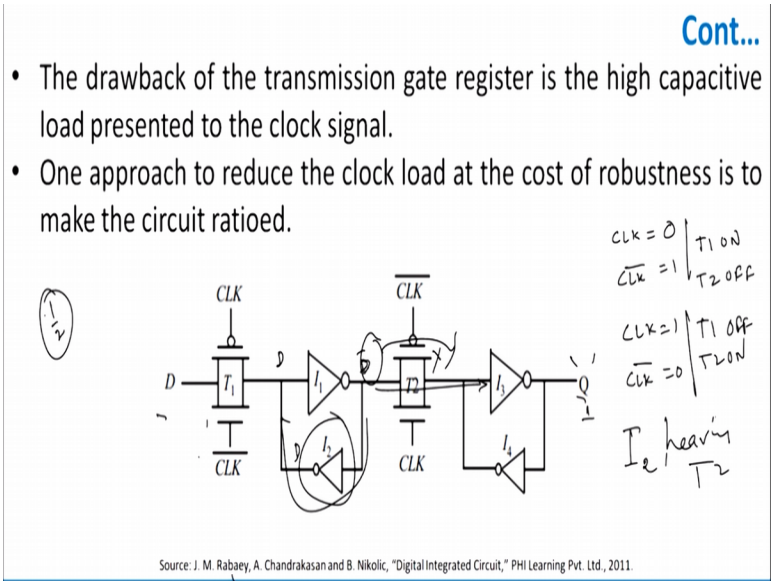
(Refer Slide Time: 20:00)



Instantaneously any data available at QM, instantaneously can be fed to Q without validating the whole time and therefore my T hold is equal to 0. This is where again in marks based design right. So remarks based design my T set base is basically 3 times inverted delay plus 1 transmission gate delay and T hold time is approximately equals to 0, because the reason being.

The whole time is 0 because once you are evaluated your data right, once you have evaluated your data at QM, you just have to wait for the next raising edge of the clock for the slave to sample the data. so that is the only delay which you get not the whole time delay available to you right.

(Refer Slide Time: 20:42)



So that is the reason we get such type of delay, this thing is there. However there is a problem here. See the problem is something like this. The problem is that if you look very carefully the clock is actually feeding 1, 2, 3, 4, 5, 6, 7, 8. So the clock is feeding 8 transistors right and therefore the clock loop is typically very high. Which means that the clock has to drive 8 transistors at each point of time, but please understand this is just a single master slave configuration.

So you can have n number of configurations so that n into 8 will be the total number of transistors which the clock has to drive right, which is quite larger quantity if you appreciate very carefully which means that, the clock load becomes very high for such a configuration because you are actually using a TG configuration there. And therefore the capacitance is also very high.

Because you are driving larger number of gates each gate will automatically have a CROX which is basically oxide capacity per unit area and therefore if you multiple with 8 number of gates the capacity will be loading on the clock will be typically very high right. And therefore there will might be a problem as such in terms of doing it. one way of one more thing which I should.

Which I sorry which I should keep you in mind is that this logic which you see in front of you is basically a non-ratio logic. Which means that it does not matter at this stage what should be the aspect ratio of T1,T2, T3 and T4 as long as they are able to pull the data or transmit the data from input to output similarly for invertors as long as they are able to invert the signals from 0 to 1.

At this stage we are not very much interested in to knowing the whole concept here right about type about delay and everything else. Just reaching the functionality. So I can afford to have a non-ratio logic here. What people gave an idea was that let us make it ratio right, let us make it ratio. And let us do one thing, let us make the feedback loop I2 and I4, so if you look very carefully this is D.

So let us suppose when clock A goes to 0 so clock bar will be equals to 1 which earn that T1 will be on state as a result D will be available here it will be D bar here but what about T2 since clock bar equals to 1 and clock equals to 0,T2 will be in off state. So when this is off this is gone, this is not coming into picture this Da will become to D bar will take a turn here and it will convert into a again D here and therefore there will be a infinite loop available to you right.

Now what you do in the next lot cycle when clock equals to 1 and clock bar equals to 0just the reverse happen T1 goes off and T2 goes on fine. And therefore whatever the value available here of the D will be sampled through T2 onto I3, I4 pair and that will come out as Q. so Q was again therefore not transparent to D at any point of time. Q and D are still non transparent right and they are therefore different.

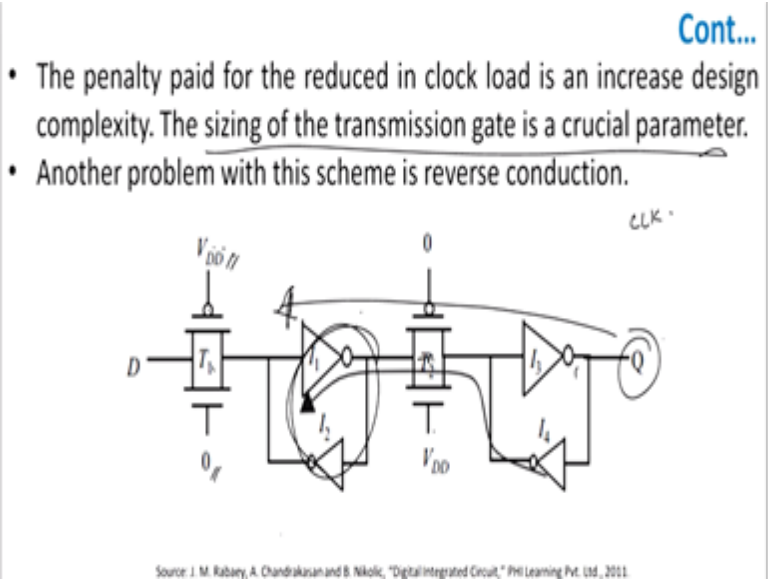
But interesting idea is that if you can keep the feedback loop I2 and I4 slightly weaker transistors compared to I1 and I3 right. Suppose if you are doing I1 and I3 right and I1 and I3 are stronger as compared to I2 and I4 which means that your forward transistors forward inverter is stronger than your backward invertor. Then, when you are actually sampling the data in the evaluation board.

Which means that in your T2 is on state then you can force the data which is here to come at this point whereas it would have been a very large transistors and then it would have been I2 would have been heavier or as would be loader than I, T2 then the data will not at all be read by T2. So I2 will drag all the data, all the voltage across itself and then you can actually there will be a feedback loop available to you fine.

So, this is the basic idea or a basic concept of a transition register right, this is TG based register. Here you are giving a so the clock load there was 8 here it is 4 so you have reduced the clock load by approximately half so in the previous case the clock load was 8 in this case

the clock load is approximately equals to half and we have achieved it by making it leisure. And making the I2 and I4 relatively less stronger than compared to I1 and I2 right.

(Refer Slide Time: 25:23)



As I discussed with you, that therefore it becomes important that the costume pay for this is basically the raising of the transmission gate is a crucial parameter right. And that is the major problem you face that you need to therefore size your transmission gate and your feedback reverse bias loop in such a manner that you're your output is stronger as compared to the feedback loop avail to the feedback loop.

Another big issue which comes into picture is what is known as the reverse conduction is something like this that let us suppose I so let us suppose that my clock is equals to 0. So clock equals to 0, clock bar equals to 1 right in clock bar equals to so what I am trying to say is if at all it might happen it that you do have a clock overlap. Which means that this ideally this this is P mass and this is your N mass so if you go back to the previous logic.

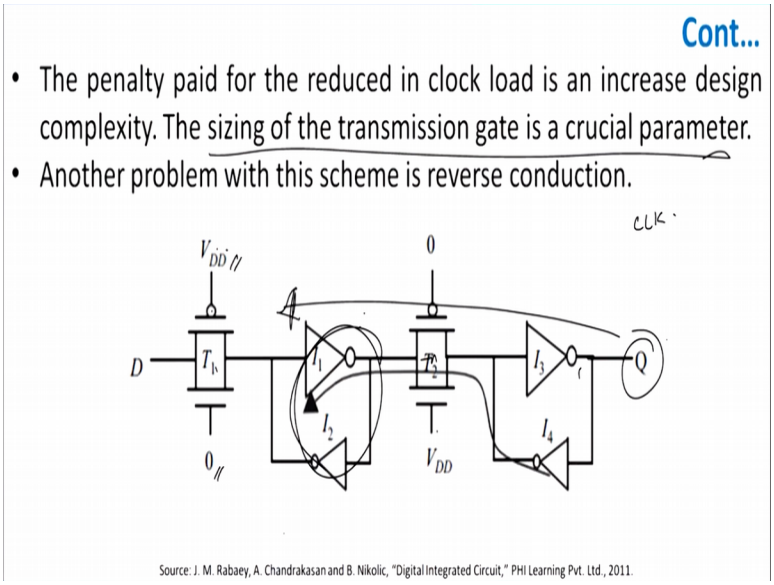
So what I am trying to tell you is that let us suppose this is clock right this, if this is VDD, this will also VDD. This will be 0**.** That is what is shown in the next slide that is VDD 00 right. If this is what is happening then T2 will be on right so T3 will be off for sure but T2 will be on when T2 is on there might be reversal from Q side plus this whole thing. So Q is the output right but the Q so if you have not made I4 less stronger as compared to I3 chances are that Q via I4 go to T2 and from T2 it goes via to this loop.

And therefore its it what is known as conduction known as reverse conduction. So you want the data to flow in the output but in certain conditions if it is a ratioed logic chances are that

the output Q end up somewhere in the I1 to I2 loop. As a result the when you are sampling the next set of data the T2 transmission gate will not be able to understand which data to sample in reality so that is the major problem of reverse conduction as far as this structure is concern right and that is the major problem area of the whole design.

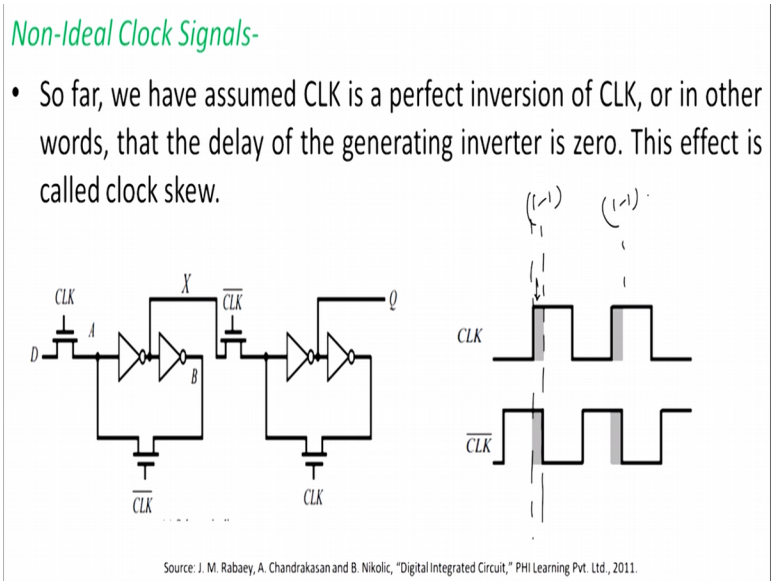
Now, so far we have all assumed that your clocks are ideal and there is a perfect inversion of clock which means that if a clock is high the lock bar is low and so on and so forth right and we have also seen that this is almost true for all the cases.

(Refer Slide Time: 27:59)



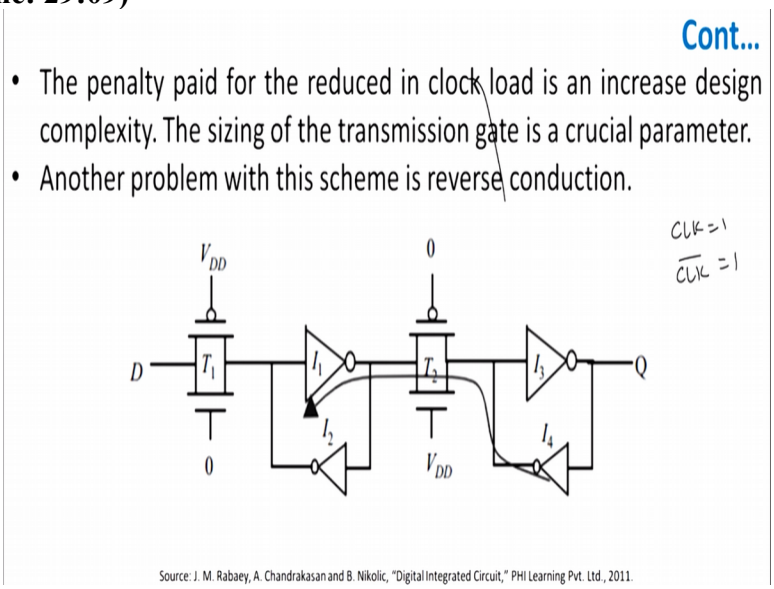
But, let us look at this point of view of actual silicon, when you run the clock across a silicon right. There are CL so on and so forth as a result the clock pulse actually shifts in time domain which means that in time domain it starts shifting to the right because of delay and so on and so forth. So it might happen see you really clock in clock bar should be opposite to each other but it might happen in certain cases that clock and clock bar, they are all overlapping with each other.

(Refer Slide Time: 28:32)



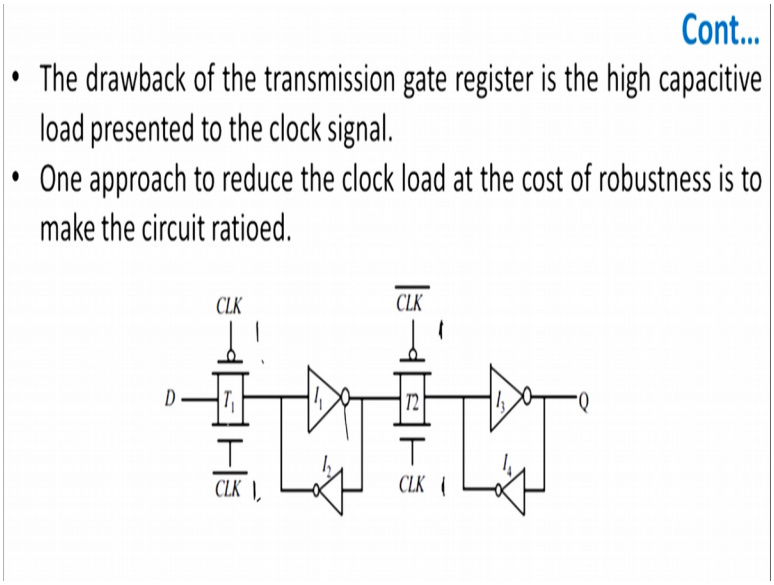
So this is the condition for overlap this is known as the 11 overlap right, so this is a 11 overlap which is here right. So this will be basically a 11 overlap therefore this is also known as clocks skew for sure but this will be basically a 11 overlap which you see in front of you. Do you ask me what is the problem, the problem is very simple and straightforward if there is a 1 overlap for example then let us see what happens what is the problem which you will face.

(Refer Slide Time: 29:09)



The problem which you will face is a if it is a 11 or overlap when which means that your 11 overlap basically means clock equals to 1. Let us suppose clock bar also equals to 1 right if both are 1 then let me go back to the previous slide again.

(Refer Slide Time: 29:26)



And if you see so if you see then if both clock and clock bar equal to 1 then this will be 1 right and clock will be also 1 this will be also 1 right. If these 2 are 1 here right and these this is 1, this is 1 and this is 1 then there might be a chance the en mas of T2 and of T1 they might get switched on at the same instance of time. And therefore you might have a chance of reverse conduction of till point D right. And that is the major problem area when you discuss about non ideal clock segment right.

Similarly if you look at this slide which is in front of you, if clock equals to 0 say clock equals to 1 then clock bar equals to 0, if clock bar equals to 0 clock 1 will clock will make 1D equals to A and A will come here, X will be close to A bar and B will be close to A. so when clock bar is A exactly equals to 1 also because of 1 to 1 overlap this A will be fed here and you will again get the same A bar and A right.

So even in the condition of overlap, this will still behave in a simple master condition right. So we will stop here now and we will discuss the next in the next fifteen to twenty minutes. Thank you very much.