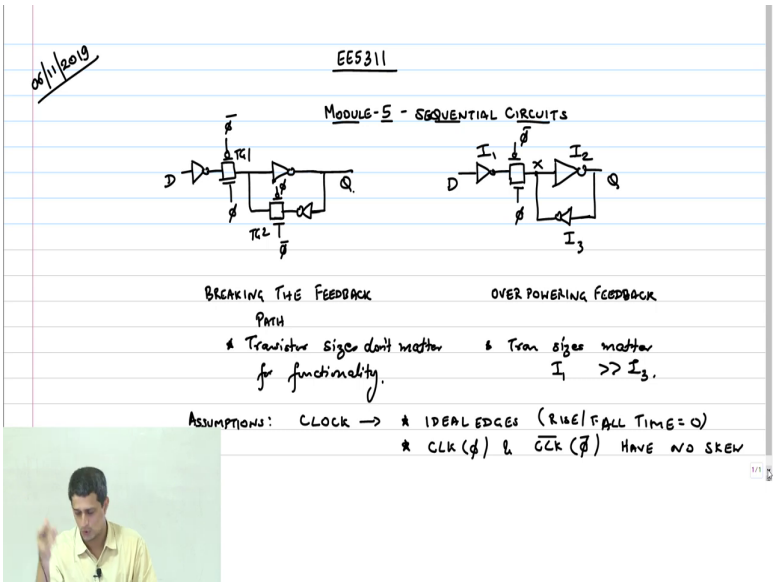
Digital IC Design
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture - 71
Clock Overlap

(Refer Slide Time: 00:25)



So, last class we looked at some circuit implementations of latches and flops, right and we could broadly classify them into two kinds right. So, we had one like this right. Let me call this phi phi bar phi bar and phi clock, right.

So, this was my D, this was my Q, then alternatively we said we could have another implementation of such a flop of such a latch right which is like this. So, we can retain this path on this curve phi, but remove this you know TG 1 and TG 2, we can remove TG 2, right. So, this was basically called breaking the feedback path and this was I last time I said it was

not breaking a feedback path, it is better known as over powering feedback, right and here the T advantage we said was transistor sizes do not matter for functionality, right.

Here transistor sizes matter, right. So, this is I 1 I 2 and I 3. I 1, I 2 should be much greater than I 3 because there will be the node x will be in contention between I 1 and I 3 when you are trying to over write a data, right. This is DQ, right.

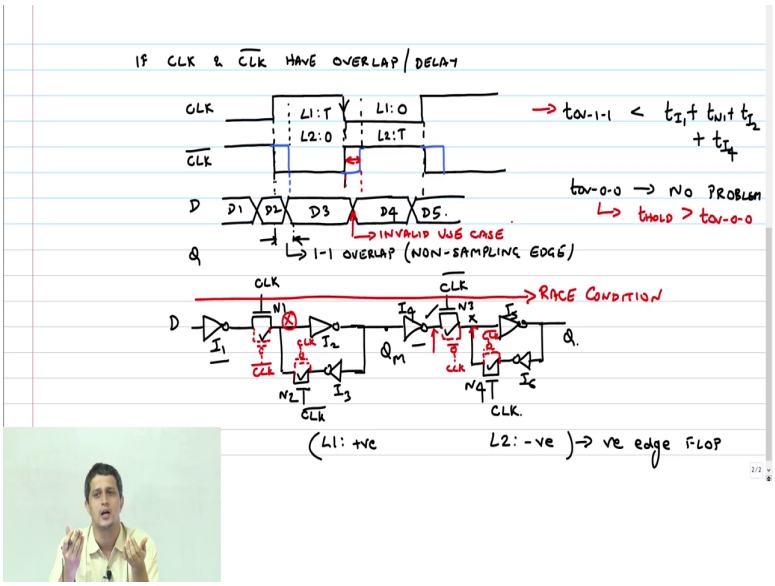
So, if you want the node x to flip, then I 1 must win at under all process conditions and therefore, I 1 must be much larger than I 3, right. The reason I am saying I 2 must also be larger is when I cascade these latches and make a flop, then I 2 becomes like I 1 for the next stage, right. So, you need that actually not true because there is an I 1 there which is available to overpower it, but if you do not put that I 1 in between there, then you might have to for example there is an inverter here, then the next stage you could choose to skip I 1 sometimes, ok.

If you are with that inversion in which case you should make sure that I 2 is also larger ok, otherwise I would say this is not absolutely necessary ok. This is where we stopped last time and we had made some assumptions about the clock, ok. One it is ideal edges right which means rise slash fall time equal to 0 in our analysis till now, then we also made another assumption that clock right or phi phi are shown here and clock bar that is phi bar have no skew. What is skew?

It means the edges are perfectly aligned right. This is the assumption that we made and based on that we got some analysis, we got some expressions for the set up time and the clock to Q delay and the whole time and all that and we found that the whole time was basically what negative or zero essentially because we said that the clock is instantaneously shutting off, ok.

So, now we will look at a simple example to understand what happens if this assumption does not hold, ok.

(Refer Slide Time: 05:50)



So, if clock and clock bar have over lapped or delayed what does this mean? So, I am still assuming that the edges are still perfect instantaneous rise, instantaneous fall ok. That we are not removing for now. This is my clock, clock bar should have ideally been like what it should have been like this. It should have ideally been like this right.

But now I am saying that this edge is moving and it is let say moving to the right, ok. Clock bar is moving to the right. So, remember all our definitions of setup, hold, time everything is with respect to the clock, not clock bar. By definition it is with respect to CLK, not CLK bar. So, therefore I am going to move clock bar around that edge, ok. So, let us assume that this moves like this. I have heard slightly exaggerated, right. So, what happens in this period, this is a period of 11 overlap.

Ideally these two clocks should have at no overlap. That means, clock and clock bar cannot be equal at any instant of time, but now there is a 1-1 overlap in that region and in this region here there is a 0-0 overlap, ok. So, what we want to do is, we want to see what happens because of these two. Let us say the whole time the whole time was 0, right. We wanted to see if that is going to change. So, for that we will consider a rather simple example where I am going to remove the PMOS transistors from my translation gates from the previous circuit, ok. So, I basically have this, ok. Size does not matter now, ok.

This is clock clock bar clock bar. So, this is QM Q right. So, and what is this? This is clock by the way ok. First of all latch L 1 and latch L 2 latch L 1 isn't a negative latch or positive latch.

It is a negative latch because so is L 1 is a negative latch or a positive latch? The positive latch because when clock is one data is shooting through the thing and getting to the output right, this is my D. So, therefore this is a positive latch and L 2 is a negative latch. So, therefore this is a negative edge triggered flip flop, ok.

So, together you get a negative edge flop ok. So, now I want to impose this condition that I just told you where I have a 0-0 overlap and 1-1 overlap, ok. So, what first of all what is the expected behavior, ok. Let me maybe I will just remove this I can use the space right ok. So, I have my data and D and this D as usual is making some transitions here. So, this is D1 D2 D3 and D4 and it eventually go to D5, ok. What is the expected output on Q? What is the expected which data should it sample?

This is a negative edged flip flop right and therefore, it is going to sample which data is going to sample D3 and it has to hold it right and yeah and the change in D2, what should happen to that when it goes to D2 from D2 to D3? I am sorry.

What should happen to that what is shown in the arrow here is a sampling edge, correct. At sampling edge you should sample what should happen if the data changes on the not sampling edge should be ignored, right.

Data should not go through that is the intent. So, that is what we want to see and we want to understand what happens here, ok. So, let us look at the case when my data changes out here which is what does this overlap.

11 overlap. So, if both clock and clock bar are 1, it means that both these pass transistors will be on right and of course, even these transistors are going to be on now for a short period, right. So, any change in my data D right like show here when it goes to D2 right at the non-sampling edge can actually raise through from the input to the output because there is a path available here like this, ok.

Now, let us put down what should happen after we overlap period in both cases, ok. What should happen here? For example, in this period what should the state of L 1 and L 2 be in terms of transparent or opaque clock is one transparent L 2 opaque. Similarly I have something here right. L 1 is what opaque and L 2 is transparent, correct.

So, now my question is there is an overlap and data can potentially shoot through right, but question is does the data always reach Q? Always is there some delay constraint that I can impose, so that it does not reach Q. That is all I need. My flop behavior is data is changing on the non-sampling edge that should not appear at output Q. That is all it can affect any internal node, right. So, you see what happens after this particular period here right in the sense L 1 is going to become transparent after the 1-1 overlap correct and L 2 will become opaque which means it is going to hold the data. So, suppose the data actually cross this flop here.

If the data came all the way till here, then you are done. It is going to sample that and hold it to the rest of the period right. So, therefore I cannot allow my data to shoot through till here, but I can allow it to shoot through till here, correct. Let me yeah because after my clock bar goes low, it is going to shut off that thing. So, as long as that feedback loop is not affected, I do not have a problem right. So, what is the delay constraint in relation to this overlap period that I can impose?

Basically what I am asking is what is the maximum 11 overlap that I can tolerate, so that this flop does not misbehave. So, you said two inverters.

So, let us know label them I 1 I 2 I 3 I 4 I 5 and I 6, then I have N 1 N 2 N 3 and N 4 yeah. So, what is my condition t overlap 1-1. So, it should be t I 1 plus t N 1 plus t I 2 plus I 4 yeah right. If I can ensure that it goes only this far, there is no problem. So, what should be the inequality between the left and right hand side t overlap 1-1 should be lesser than this.

If I ensure this, then there is no problem, ok. Remember now this overlap condition is a little complicated in the sense that if there is a violation, then my flop is gone. So, it is your job as a designer to ensure that there is sufficient delay from the input to the output, so that you do not have this race condition, ok.

Now, yeah?

Correct.

No, but how will it cross I 1 and I 2 without crossing I 1 and N 1?

No be careful this is a non-sampling edge, this is not the sampling. The sampling edge I agree with you set up time will ensure that data is stable, then data has to go pass to only that from you know I 2 to I 4. This is the non-sampling. So, data should not even enter actually. Be careful about that. That is that is the whole point right let me. So, here this is the non-sampling edge, ok.

So, you have to so now you think about it. Suppose I design a flop without this I 1 and I 4, it is possible right. I mean data will be inverted t N Q N may be inverted, but that is I can always have some flop like that right. Then remember that you are actually reducing this delay that is available to you. If I have more invertors on the path, then I can tolerate more clock overlap right. If you reduce that, then you are running into the risk of reducing that

overlap time and you have an issue. There you have to be very careful about that, ok. So, that is.

So, this is called a race condition, ok. Now what happens on the 0 yeah?

No, what is the t N 2 or N 3 N 3.

So, what happens, see the thing is this guy is in the there is a feedback inverters on the other side. Now if this capacitor after N 3 charges slightly about the trip point, then that feedback inverter is just going to hold that value for ever. So, to be careful it is better to cut it off before.

Yeah that is all. This is you can be sure if you are if you impose this condition here that is all beyond that there is a risk ok. You can tell the designer, you can tell the consumer of the flop that look do not go beyond this overlap time. For sure it will work, but beyond that there is a risk, but in reality you are right there is as I told you the edges are not ideal and all of this is not ideal, right. In that case the number will be between considering N 3 and not considering N 3, ok.

Now what happens in the 0-0 overlap case? Suppose what happens first of all to the NMOS transistors when there is a 0-0 overlap, all of them are off which means that there is no path available from D to Q. Forget D to Q even from D to Q M, there is no path and therefore, you really do not have a problem in this flop, right. T overlap 00, no problem yeah.

Where?

Correct, now if D changes, no you are talking about this transition right, what I marked here D3 to D4, correct that the sampling edge you are talking about that transition, but I do not D 4 to go through I want D 3 to go through D 4 has change setup time before the sampling edge and that is what should go out to the end, not D 4. The moment that is why I am saying

everything is the reference to the clock, not clock bar. After that clock bar sampling edge I mean after the clock sampling edge, any change should be ignored right.

No in this configuration there is no problem because the moment I go past the sampling edge, all my NMOS transistors are off feedback path and direct conduction path. Yeah.

No first of all data changing at the negative edge has no meaning here because it is an instantaneous thing right. So, it changes let us say just after that there is no problem it will I am saying under the assumption that the clock edges are instantaneous, all clocks turn off and therefore, there is no way D can enter QM. Forget Q.

Wait I know I will come to that. That is that is not in this configuration, ok. I totally understand what you are saying, but in this configuration it does not hold, ok. So, let us now answer this question yeah.

So, when we talk about these overlaps and all these, it cannot be a significant fraction of the clock period right definition. These overlaps are very small deals, otherwise you have a problem in the system right. Clock, overlap, jitter, skew all of these things are very small numbers compared to the clock period otherwise your system is not going to function anyway ok. That is a corner case, case that we do not have to worry about, ok.

So let us know. So, what is so magical about this flop, right? The 0-0 overlap there is no issue how come. So, what other configuration do you think will have a problem in the 0-0 overlap? All I have to do put a PMOS transistor in here right in here and here and all that and have these transistors conduct even in the 0-0 case. That is all right. With the other PMOS case end connected to the appropriate clock polarities clearly if clock and clock bar are now 0-0, then instead of the NMOS all the PMOS transistors will be turned on and therefore, this is a condition which is very identical to what we saw earlier on the non-sampling edge were there was a race through condition was possible.

That means, the change after D3 right which is basically this guy this change after the sampling edge D3 to D4 can potentially raise through ok. Now you have to carefully see what

happens, how much of delay can we tolerate there. Let us say right like the last time I am trying to see if there is a condition like this that we can arrive at let us say that the data just got past the, past this transmission gate into this node x. Let us say just got in there in what mode is L 1 after that after the overlap period.

Yeah L 1 is opaque right which means that this feedback part is going to hold that value what happens to L 2. It is transparent. So, is the data sneaks pass that transmission gate and just gets to node x data will get sampled all the way to Q and you are gone. So, I have I can laterally tolerate no delay in this case. So, what I am asking is like the last time we said that you know the data can come all the way till here, but not till here and we imposed some conditions for the overlap time. Is there a similar condition I can impose here is what I am asking.

For that you also analyze and see what happens after the all law period, ok. So, this is my 0-0 overlap period that I am considering. What is the state of L 1 and L 2 after overlap? L 1 is becoming opaque L 2 is becoming transparent, therefore is the data gets to node x if the change in D affects node x right. That is x now becomes D4, then data can go through all the way to Q.

Because after my overlap period this feedback path is going to hold that value the L 2 is now going to be transparent. Therefore, data can go through all the way after the overlap period and therefore, I can tolerate no delay. Actually in this case I cannot be putting more invertors, nothing I can handle. That kind of a delay data should not go past that transmission gate. So, how do you handle it if I put a PMOS?

No same thing I am saying you connect this polarity to clock bar and this to clock right and this again to clock clock bar. No polarity change. I am just making a past transistors into a transmission gate. That is all right. So, what do I do? At least earlier I had a couple of inverters and past transistors where I could absorb that delay and so if I want it for example more delay to handle the larger overlap, I have to maybe add more invertors in this path for the 11 overlap right. What do I do in 0-0 overlap?

No. So, again for this configuration what do I do?

Because whether it is a NMOS or transmission gate that comes from a different considerations whether that is a unacceptable flop or not right, it has its limitations right. It cannot be used everywhere, but in general what do you do? So, that is where this whole condition comes in.

I can tell the consumer of this flop that look you need to hold the data after my sampling edge at least for some time. What is that time? It is this clock overlap time if you will tell the consumer that if you change the data before this hold violation hold time, then I cannot guarantee that this flop will work correctly. So, this you basically say impose it to the hold time and what should be hold time should basically be greater than t overlap of 0-0. So, if you impose the condition that is larger than the worst case overlap of 0-0 that you would see, then the consumer of this flop this becomes an invalid condition, invalid use case. Hence I do not have to worry that is the consumer's problem, ok. Any questions here?

Yes, this is the minimum hold time that you will see in a flop and the hold time origin is basically clock overlap, ok.