Digital IC Design

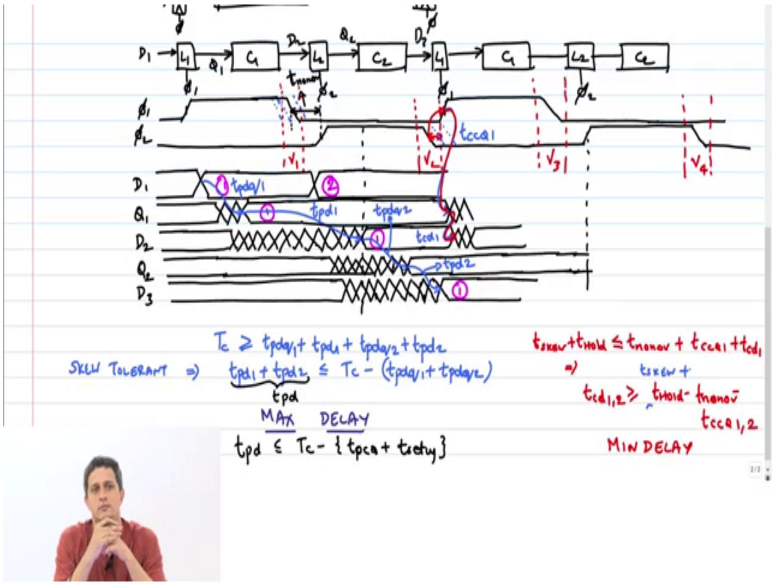
Prof. Janakiraman Viraraghavan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 77
Latch - Time Analysis with Skew

So we were discussing latches right and we were discussing the concept of; we started the idea of time borrowing, but we really did not derive any condition for that. But before that, let us just revisit what we did last class but this time with a slightly different clock wave form.

Let us assume that the non overlap period is actually very small; because that is why it was not evident again that there would be a whole time violation. Because there is a long gap right non overlapping clocks give you that advantage you will never have a whole time violation because that non overlapping time is large. But suppose that were not the case then we will see what happens today right.

(Refer Slide Time: 01:01)



$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}$

SKEW TOLERANT $\Rightarrow$ $\underbrace{t_{pd1} + t_{pd2}}_{t_{pd}} \leq T_c - (t_{pdq1} + t_{pdq2})$

MAX DELAY

$t_{pd} \leq T_c - \{t_{pcq} + t_{setup}\}$

$t_{skew} + t_{hold} \leq t_{nonov} + t_{ccq1} + t_{cd1}$

$\Rightarrow$ $t_{cd1,2} \geq t_{skew} + t_{hold} - t_{nonov} - t_{ccq1,2}$

MIN DELAY

So, let us just revisit the latches max delay and min delay constraints ok. So, what is my constraint let me just go to the new page; because I need more space I have latch L 1, combinational block C 1, L 2, combinational block C 2. Now instead of going to L 3 I am going to bring back L 1 ok. Physically in space this is not a L 1 I am actually unrolling this in time.

So, that is why I am saying you look at L 1 again at that particular edge ok; ideally I would have had L 1, C 1, L 2, C 2 then L 3, C 3, L 4, C 4 there will be different blocks right physically in space. But now I am trying to consider what happens in to L 1 at a different time instant ok. So, I am unrolling this in time not in space ok. So, this will be back to C 1 and L 2, C 2 ok.

So, my clock phi 1, phi 2, phi 1, phi 2 ok. So, let us look at phi 1 here I am going to have a rather long period so that is the non overlap period is very small just. So, that you get a clear picture of what why there is going to be a whole time violation phi 2 again both latches are positive latches ok.

So, what is going to happen here? My this is going to rise somewhere here right and so on right. So, now, what is my non overlap period? It is basically this period right, this is t non overlap ok. And of course, from this raising edge to this raising edge it is my clock period itself phi 1 raising edge to the next raising edge is the clock period itself ok.

So, now what happens is, I have data D 1, Q 1, D 2, Q 2 there is this D 3 and now I am coming back to L 1 ok. So, physically that is not the case so, I will not worry about labeling that for now ok, we will come back to it when we deal with the min delay constraint ok. So, now, what is going to happen I am going to have a change in D 1 that needs to be captured and before we proceed let us mark our violation window very clearly right.

This is my violation window for phi 1; falling edge is when it is going to sample the data and it has to hold that data until the next raising edge right until the next data comes that is the; that is the point; if it does not hold it there is a violation right. Similarly for phi 2 this is my violation window ok, phi 1 again I have a violation window here phi 2 this is my violation window ok.

So, I am going to have a change in D 1 like this, D 1 it has changed and it has given rise to event 1 that all these blocks need to process. So, that event has 1 changed 1 has changed L 1 will capture it hold it C 1 will process that then it will go D 2; D 2 will then get sampled by L 2 it will hold it and that has to proceed that event 1 has to be processed by each and every latch and combination block appropriately ok.

So, now what will happen to Q 1? If you look at Q 1 where will it change or what is a delay it will incur? Pdq right, because I am saying that in a latch, data does not have to change exactly before the rising edge it can change even afterwards, remember the latch has to hold even 1 right and just hold it till the next raising edge for that that the data has to settle.

Before event 1 has to settle before this violation window; that is all I will call the violation windows v 1, v 2, v 3, v 4; event 1 has to settle before violation window v 1 then it gets sampled correctly there is no problem ok. So, what will happen to Q 1? Q 1 is there will be some small variation of course, or some small contamination delay right now we are looking at the propagation delay.

So, I am not going to mark the contamination delays right. So, this guy D to Q right this is t what t p D Q tpdq ok. Then what happens to D 2? That is going to again it will start with some contamination delay let me start that delay very early right; there will be some long thing before the propagation delay actually allows you to settle ok.

This is more like a realistic picture the contamination delay can go through C 1 very quickly right but the propagation delay may still be long and the time to settle may take it well into phi 2 ok. So, this is what D 2 right now what happens to Q 2? So, it has gone into phi 2; that delay it will increase is what tpdq again right. So, Q 2 is and of course, this is event 1.

Now, when will the contamination delay start for Q 2? Can I start here? What is the contamination delay that Q 2 will see? It has to wait until the clock turns on. So, therefore, contamination delay can come only from clock to Q right they cannot come from data to Q; because the clock is not turned on any change here is not going to effect the data there correct are all of you with me here? That is why for a max delay we are considering the data to Q delay but for hold violation we are considering the clock to Q delay very important this point so, it cannot be this.

As soon as the clock changes here, you will find that this can start changing like this right and it will settle after some delay ok. Now what are the delays here? From this is what t p d 1 right for combinational blocks C 1 it is t p d 1 correct. From here to here again what is it?

From D 2 to Q 2 change is what delay is t sorry, this is tpdq again right. So, this is tpdq 1 pp or I will just call it tpdq identical flops ok; same delays now what will happen to t D 3?

So, actually now I am not going to look at D 3 here I am going to look at D 1 right. So, what do I want? So first of all what is the max delay constraint here? Assuming that I am not going to borrow time what is time borrowing? If you look at this D 2 right the change has happened well into the rising you know the on time of phi 2 ok. This guy the change here there has been a significant amount of time borrowing that has happened there right.

You are allowing D 2 to settle after the clock has gone high that is called time borrowing ok. We will see what the conditions should be how much of time I can borrow is also 1 condition. So, we look at 1 that is right how much of time can you borrow in a flop based system? 0, its a hard edge if you miss that edge you have to wait for the next cycle right, that is the whole advantage of going from a flop based system to a latch based system because I can borrow time right. So, we will see some advantages of this very clearly later ok.

So, assuming that this has to finish within the clock cycle t c itself what should be the condition now? So, t sorry, D 3 I have to look at D 3 I am sorry D 3. So, where is what will happen to D 3 now? So, now, D 3 what is the contamination delay with respect to what will it start? As soon as Q 2 start changing it can cause a because its a combination clock it has nothing to do with the block right. So, therefore, this will start changing somewhere here right and so, this is what p t p d 2 correct.

So, what is the constraint for the max delay? T p d 1 no be very careful where is the violation window now? Huh v 2 no hold on v 2 is being sampled by phi 2 correct. Now I am looking at psi 1 right D 3 it is now going to be sampled by phi 1. So, therefore, there is no violation window it is very far away; that is why I am drawing this waveform again specifically if you draw it with very large non overlap period these things may not be obvious ok.

So, therefore, there is really no violation here violation window there is a long time if I have to finish within that cycle; if I can borrow across cycles then you are right. I had to finish it I had to finish before the violation window v 3 not v 2 also ok. So, therefore, I have my clock period T c and then what should happen I have my delays t p d q plus t p d 1 plus t p d q right I can call it 1 and 2 actually not d q right plus t p d 2 and what is the inequality greater than or equal to clock period should be greater than this; if clock is slow no problem you can accommodate it right.

Implies t p d 1 plus t p d 2 should be less than or equal to T c minus tpd q 1 plus t p d q 2 right.

Student: (Refer Time: 16:17).

No you are right I am saying of course, if you can borrow time that is what I said, borrowing time from phi 2 is time borrowing across half cycle; borrowing time across a half cycle is always allowed. So, that is why we are not worried about resting within the half cycles. But time sorry time borrowing across cycles is prohibited in some cases I will show you next; that is why I am first deriving the condition saying.

If I have to finish with in the same cycle then this is the condition; that is the amount of time borrowing yes, I will come do that next ok. So, now when can D 1 change? Let us assume that D 1 is coming from a primary input where I have a control I will say that ok; you should not change data in a certain period. So, if I want event 1 to be captured ok, this is event 1.

If I want event 1 to be captured, when can D 1 now change? Immediately after now what tell me which violation window? After v 1, the earliest time that v 1 can change successfully, without affecting the operation is after v 1 has been sampled correctly to Q 1; which means that v 1 can now change only somewhere here. So, if you change like this then this is event 2 ok.

Now, what should happen to this, let us look at D 1 has come through there. So, what is the constraint on D 2 how long should D 2 be stable right what is the job of Q 2 first of all? After phi 2 falls, data has to be sampled and it should be held until the next rising at the phi 2 correct. The intent is Q 2 should remain at event 1 until this period correct is that right correct right?.

So, now in order for that to happen, D 2 should be stable for how long after which violation window? So, D 2 should be stable until after v 2 violation window. Now let us see what happens when event 2 comes in on D 1 right after the violation window v 1 no problem so that has been taken care of.

Now, because D 2 came here you look at phi 1 again ok, here because that is only when phi 1 goes high again will the change in D 1 be allowed to go through to Q 1 correct. So, as soon as phi 1 goes high, I am going to have a contamination delay in 1 in Q 1 right. So, therefore, you will look at Q 1 now, as soon as phi 1 goes high I will have a contamination delay this is what t which one contamination t c c Q 1 correct.

Now as soon as there is a contamination delay on Q 1, it can reach D 2 because it is a combination block correct. So, what will happen to D 2? D 2 will start changing like this right, here these are actually very small delay that is why I am showing it trying to show it I will remove this maybe I will put that here; t p d q 2 this what is this delay? From Q 1 to t 2 t c d 1 correct.

Now, what is my constraint on D 2; D 2 has changed what did we say or when did we say that D 2 is allowed to change so that there is no violation after violation window v 2 right. So, this is what happens, if data changes and the overlap period is very very small non overlap period is very small I am sorry right. If that happens, when this data can raise through from D 1 to Q 1 through C 1 and contaminated data that was supposed to be held by D 2 in the earliest cycle; are you with me? So, what is the constraint for this to work correctly?

Student: (Refer Time: 22:39).

T hold is with respect to which clock phi to now right. So, from here I have a hold window right; this is my I think that is red this is my old window correct. Now what does what are the delays that I am going to incur? First is the non overlap from phi 2 to phi 1 right; because you remember that now the changes that you see in Q 1 and D 2 are because of the clock phi 1 not phi 2. So, therefore, I have to now go from this edge through this delay right then through this delay.

So, I have to count this delay as well what is that delay? That is the t 1 overlap. So, therefore, my min delay constraint is t non overlap plus t what c c Q 1 plus t c d 1 and I have a t hold constraint here what is the inequality less than or equal ok. Violation window v 2 should be small enough or the data change in D 2 should happen after that right which implies my t c d 1 should be greater than or equal to t hold minus t non overlap minus t c c Q 1 correct.

This is my min delay constraint this is max delay this is my min delay ok. Now this t c d 1 that I said, whatever discussion we did we have to do it even for C 2; the same discussion will hold even for the outputs D 3 right, D 2 should not raise through and cause a change there everything. So, for every latch I have to do this correct. So, therefore, I will simply say t c d 1 comma 2 t hold t overlap 1 comma 2 t hold minus t non overlap minus t c c q 1 2 right. Same condition is going to hold there as well just the delays might be different.

So, last time I left you with 1 question, I said that max delay constraint is 1 constraint if you look at this it is t p d t p d 1 plus t p d 2 less than or equal to T c minus some group propagation delays right. So, we got 1 constraint for max delay and 1 constraint each for min delay in within 1 cycle that is for the 2 latches correct. A flop on the other hand has only 2 constraints; 1 max delay, 1 min delay this is for a latch. How does a flop look by the way the condition t p d it should be less than or equal to T c minus t c no P C Q plus t set up correct.

You remember this condition for the flops, this is what we had because we have hard edges you have to set up before that edges and all that. If you look at the max delay constraint its not very different; because t p d is the net suppose I had implemented this system with a flop what would I have done? I would have put a flop f 1 here right and then combine the block C 1 plus C 2 and then put another flop here.

So, this would have been phi. So, if you look at the constraints here, its not very different t p d is less than or equal to t C minus t p c q plus t set up for a latch it is. This any way is actually t p d and the T c sequential sequencing over ride is what t p d q plus 1 plus t p d q 2; if you look at it if you remember what we did derived earlier, the t p c q and t set up are nothing but the t p d q and the t p d q I mean of L 1 and L 2 separately.

So, this is a not very different, the max delay constraints are almost identical it is clock period minus 2 latch delays in a flop we call it t p c q and t setup in the latch it is just p d q 1 and p d q 2. So, its not very different, the min delay constraint; however is different, we are getting 2 conditions as opposed to a single condition which was t c d should be greater than t hold minus 1; t c c q right that is what we got there.

But where did that third condition vanish because a flop after all is just 2 back to back latches. We have taken that latch out and put it somewhere in between split that combinational block which was C 1 plus C 2 make it C 1 C 2 and inserted that L 2 here. how did that third condition come in suddenly which was not that earlier?

Student: (Refer Time: 29:11).

Correct that is because you have control of C 1 and C 2.

Student: (Refer Time: 29:35).

No my question is did the condition really get removed at all?

Student: (Refer Time: 29:45).

Ah.

Student: (Refer Time: 29:51).

Ah.

Student: (Refer Time: 29:53).

Ah.

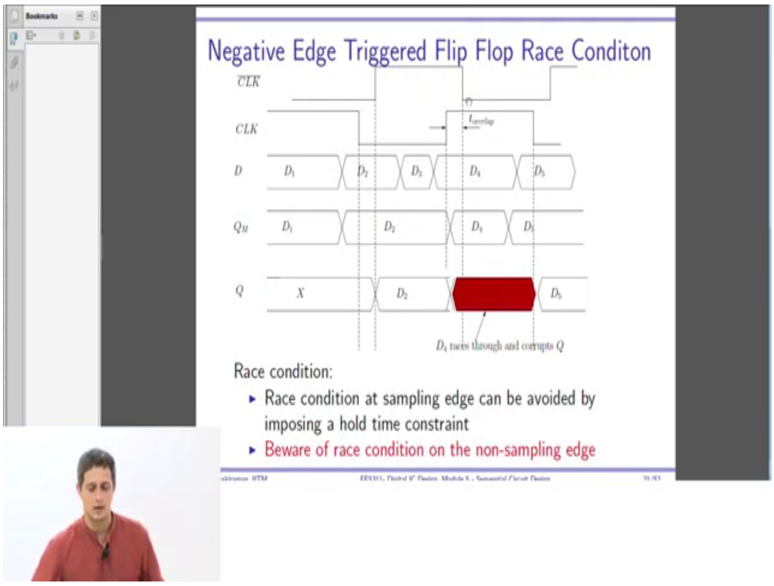
Student: (Refer Time: 29:55).

T hold is ok. So, what I am asking is basically now you have in the latch system, I have 1 min delay constraint for latch per latch; in the flop there magically seems to be only 1 min delay constraints for the input flop input latch L 1 where did the condition of L 2 go? Well, the answer is the condition did not vanish.

If you remember that condition was a function of the delay of the latch inverters you remember when we did this old violation analysis. The one one overlap which happened on the non sampling edge, you had to ensure that that delay was less than the propagation delay from input to that second latch. And if you violated that like your flop is gone you will have a old violation or a waste condition. So, that condition is implicitly there in the flop design.

The difference in a latch based design is, I have control now over within that delay; because I can design C 1 and C 2 to make sure that delay violation does not happen. In the flop if you had that violation, then you are gone right let me show you that right we had that you see this.

(Refer Slide Time: 31:43)



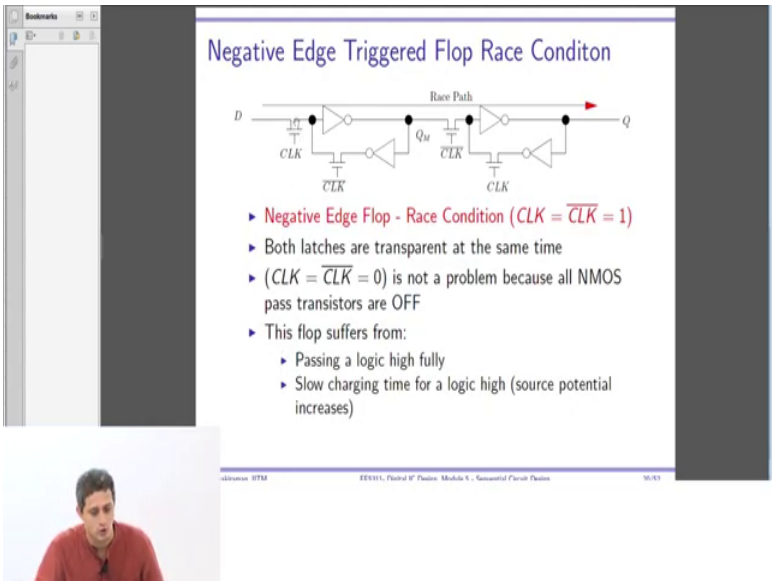
Negative Edge Triggered Flip Flop Race Conditon

Race condition:

- Race condition at sampling edge can be avoided by imposing a hold time constraint
- Beware of race condition on the non-sampling edge

This was the non sampling edge this is a negative thing right; this non sampling edge because of the overlap had to have some constraints there. And what was the constraint it basically said that the overlap period should be smaller than the delay of going through this, pass transistor through this, inverter through this, pass transistor. And know you are getting there that was an implicit condition in the flop design.

(Refer Slide Time: 32:09)



Negative Edge Triggered Flop Race Conditon

- Negative Edge Flop - Race Condition ($CLK = \overline{CLK} = 1$)
- Both latches are transparent at the same time
- ($CLK = \overline{CLK} = 0$) is not a problem because all NMOS pass transistors are OFF
- This flop suffers from:
  - Passing a logic high fully
  - Slow charging time for a logic high (source potential increases)

So, its not with the condition vanish, the condition is implicitly there. There are still three conditions but one is one is something a designer does not have control over it has to be taken care of at design time or the flop itself. If I give you the flop that internal race condition constraint is already fixed you can do nothing about it. Whereas, in a latch base system, I can do something about it because C 1 and C 2 are still in my control clear.

Please think about this its a subtle point think about it and you see what happens ok. Now what happens if I have clocks Q to these two conditions right what happens if I have clocks Q right I am going to have this sampling edge right. So, what is will the max delay constraint get affected will the max delay constraint get affected if I have clocks Q?

So, you might think that if I want to finish it within that cycle, then it will get affected; because the clock edge moves right. But I will show you later that that is not a constraint because if I want to finish it within 1 cycle its only when I have feedback; which means that is no skew because I know not dealing with the same flop again I will show you where that thing is not.

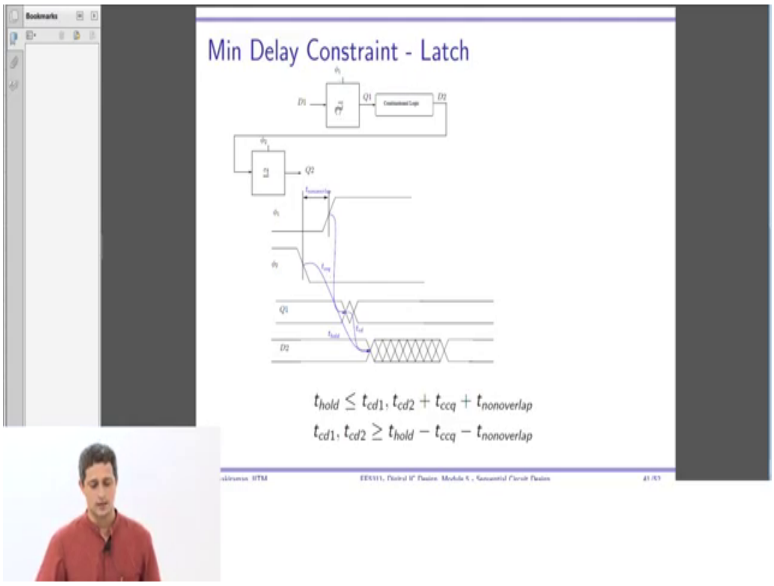
So, in general this is skew tolerant, the reason is if I have a subsequent block L 3 and L 4 and so on, then technically I can borrow some on the neighboring clock period also. So, there is no need that it has to finish in that one time period ok. If I can borrow, so therefore, even if that edge moves a little bit I am still ok. So, in general latches are skew tolerant for max delay; however, min delay that is not true because like I showed you last class.

This will become q plus this why because, now my hold constraint is going to happen with respect to the clock that moves this way; psi 2 will move to the right little bit and therefore, I have to ensure that the whole constraint is with respect to the shifted clock. So, t hold plus t skew will come in there and it will alter all this condition here.

So, this will become t skew plus t hold plus whatever, no again see the thing is t non overlap is a is a design parameter these skew is that comes in post because of manufacturing. So, you cannot combine them because the thing is t skew is variable ok; in the sense t non overlap is fixed for all clocks everywhere in the system.

Whereas t skew the worst case right is that actually with the random variable on this part of the chip it maybe some value and that part of the chip some other. So, I cannot combine it with the design parameter which is supposed to be a fixed number. So, how much of time can I borrow? How much of time can I borrow? Before that by the way let me show you this. The book textbook Western Harris discusses this min delay constraints slightly differently look at this.

(Refer Slide Time: 36:43)



Min Delay Constraint - Latch

$$t_{hold} \leq t_{cd1}, t_{cd2} + t_{ccq} + t_{nonoverlap}$$

$$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{nonoverlap}$$

EE5311- Digital IC Design, Module 5 - Sequential Circuit Design

If you look at the diagram here the way the book does it this time is what is represented here. So, if L 1 is ahead of L 2 it means I am looking at the previous cycle of L 2; that is why my L 1 and L 2 is not drawn in a straight line. Because that represents the physical picture L 1 and L 2 are going like this L 1, L 2, L 3, L 4 will go like that. So, time also proceeds in the same direction.

The reason he has sort of brought it back like this is to show that they are both dealing with the same event; I unrolled it in time and drew it on the same you know on a continuous thing that is why. Whereas the textbook has drawn it like this and my slides also has it like this. So, if you unroll this in time, then this guy will become straight here that is also that is how you have tie the what I have done in class and what is on the slides and in the textbook ok.

This is just a notation and a style of showing it ok, both are equally valid. This unrolling seems more easy to explain, that is why I am doing it like this.