VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 10

Lecture 49

Time Borrowing in Latch

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about how to do time borrowing in latches. The content of this lecture includes time borrowing in latches. There are two types of latch such as two-phase latch and pulse latch. So, we will discuss how we can do time borrowing in both the cases. Then we will go into some of the examples of time borrowing. In case of a flip-flop based system, so what happens? Let us say we will discuss about the flip-flop based system. So, let us say I have a flip-flop, I have another flip-flop and clock is given to both of them. This is some combinational logic is there. So, this is your D1, Q1, D2, Q2 and these are clock. So, in case of a flip-flop based system, this is the combinational logic. Your data is captured on the rising edge of the clock, if it is a positive edge triggered flip-flop. So, your data D2 should reach before the setup time of the flip-flop. So, your D2 should be available before the setup time of the flip-flop. Then if the data is reaching before the setup time of the flip-flop, it will be captured properly by the flip-flop 2.

This is flip-flop 1, this is flip-flop 2. If the data D2 will reaches before the setup time of the flip-flop 2, then it will be captured by the flip-flop 2 properly. Then there are two things comes into account. If it comes earlier than the setup time of the flip-flop, then data has to wait till the clock rises. So, it is wasting some amount of time, but if the data comes after the setup time of the flip-flop, then the system fails. So, the clock rising edge acts as a hard edge for the data to pass from the input to the output of the flip-flop 2. So, this is the problem with a flip-flop based system. So, how can that can be solved with a with a latch based system? Okay. So, you have, let us say this is a latch 1 and this is a latch 2. There is some combinational logic is there and you have clock phi 1 and phi 2 is going to L1 and L2.

Then what is happening is that if your combinational logic is larger than the actual time period, it will go to the outside of latch 2. Okay. So, this is let us say D1 and this is Q1, this is D2 and this is Q2. So, ideally your data D2 should come before the rising edge of the latch 2. However, if the data comes late, still since the latch 2 is transparent in the positive half cycle, so it can pass the data D2 to Q2. So, there is no hard edge unlike flip-flop. So, latch based system is more flexible. It is more flexible. So, it is a flexible system. And here in the flip-flop based system is very highly constrained system. If your data is not meeting the setup time, then the system will fail. So, this is the main difference between a flip-flop based system and a latch based system. So, in case of a latch based system, it allows to borrow time from the next stage. So, how much time it can borrow? How can you do that calculation? We will discuss in this slide. So, let us say I have a latch 1, then I have a combinational logic, then I have a latch 2.

This one is phi 1, this one is phi 2, then this one is D1, this is Q1, this is D2 and Q2. So, basically, I will draw the phi 1 and phi 2. So, the phi 1, let us say this phi 1 is this one, this is phi 1 and the phi 2. Now, this edge to this edge is your T-clock, this is your T-clock. Now, here to here, this edge to this edge, this time is T-clock by 2. Now, what happens is that, let us say your D2 takes more time to evaluate. So, what it will do? It should ideally come before the rising edge of this phi 2. But let us say it takes more time to reach D2 because of more combinational delay in this case. So, what D2 will do? It will take more time. So, let us say I have the D2 and it is reaching. I am calculating what is the worst case time boring is happening. So, here I am taking and D2 is reaching inside the positive half cycle just before the set of time of the latch 2. So, what is happening? It is coming late and it is reaching before the set of time. So, this time is basically your set of time of latch 2, T set of latch 2. Now, what are the things with us? We have basically this one. This much of time is called my T borrow. So, now what are the things there? So, this half is already T clock by 2 and obviously this rest this will also be T clock by 2. This is also T clock by 2. So, in this T clock by 2 what are the operations are happening? So, in this T clock by 2 you have T clock by 2 should be at least greater than equals to T borrow from here to here this one plus T setup this one plus basically this one T non-overlap. This is called T non-overlap.

So, this is T non-overlap. So, what is happening is that I have half clock period is given to me in that I am doing three things. One is I am borrowing some amount of time, I am satisfying my setup requirement, I have some non-overlap time. So, these three combinedly constitute my T clock by 2. So, this is my T clock by 2.

$$\frac{T_{clk}}{2} \geq t_{borrow} + t_{setup} + t_{non\_overlap}$$

So, now if I solve this equation I can find what is the maximum time I can borrow in this case. So, the time the maximum time I can borrow T borrow should be less than equals to T clock by 2 minus T setup plus T non-overlap. So, this is my final equation.

$$t_{borrow} \leq \frac{T_{clk}}{2} - (t_{setup} + t_{non\_overlap})$$

This is the maximum time I can borrow in case of a latch based system. So, you can see here also. So, we are subtracting these two parameter T non-overlap and T setup from your T clock by 2 to find the maximum time I can borrow from this latch based system. With this background we will do some more analysis and examples. So, if you look into this pulse or the clock signal there are basically when that there is a timing violation in a latch we are discussing in this slide. So, what I told you when there is a positive half cycle your latch is transparent. So, your latch is transparent. So, what it does whenever it is transparent it pass the D input to the Q with a delay of T D to Q. So, what is the opening edge? So, this is my this rising edge is my opening edge. And what is the closing edge? This is my closing edge. So, this window is very important for me. Why this window is important for me? In the previous slide if D 2 is reaching earlier than the rising edge I have positive slack. If data D 2 arrives to the latch 2 earlier than the rising edge of the clock then you have a positive slack. And if your data arrive inside the positive half cycle your latch is transparent during that time your slack is 0. And if your data arrives after the falling edge of the clock falls then we have a negative slack. Means negative slack means that your system may fail. If your data comes after the falling edge of the clock falls then your system will not satisfy your set of requirement and it will fail.

With this background we will go to the next slide where you can see this we have this phi 1 and we have phi 2 which is opposite in phase phi 1 and phi 2 are opposite in phase. Now if you can see there are three types of borrowings are there. So, what are those borrowing? Three types of time borrowing is there. So, first one is time borrowing across half cycle. Then the second type is time borrowing across stages. Then the third category is called time borrowing in a loop or a cycle. In a loop or cycle. So, if you can see in this diagram that the first category is this one. So, you have phi 1 and phi 2. So, it is taking some borrowing from the next half cycle, but all the combinational delay is evaluated in one clock cycle.

This is called across half cycle, but it is not borrowing across stages. But when we are going to time borrowing across stages means you are borrowing from one clock cycle to the other clock cycle. So, this is called time borrowing across stages. Then the third category is that time borrowing in a loop. For example, if you can see here we have a line here which is going back. So, what it says that we need to satisfy the set of time requirement of this latch while doing the time borrowing. So, set of time requirement of this latch need to be considered while doing the time borrowing whenever you have a feedback loop. So, with this background I will discuss one-one example for each of the types. So, let us start with time borrowing across half cycle. So, now we are discussing time borrowing across half cycle.

Let us draw the basically circuit first, then we will do the timing diagram. So, let us say this is your latch 1, then I have a combinational delay, combinational logic 1, then I have basically latch 2, then I have combinational logic 2, then I have latch 3. So, this is given phi 1, this is given phi 2, and this is given phi 1. So, now this delay of this one is 8 nanosecond. We need to assume that delay of this combinational logic is 8 nanosecond and this is 2 nanosecond and my time period is 10 nanosecond. My t lock, t clock in this case is assumed to be 10 nanosecond. So, let us draw the timing diagram and check that how much time borrowing is there. So, here if you can see, I have a phi 1, this is my phi 1, this is my phi 2. So, for simplicity I am assuming t non-overlap is 0. So, now for simplicity t d to q max equals to 0.

So, what is happening? So, since d to q is 0, so I have this is my d 1 pin and this is q 1, d 2 pin, this is q 2, d 3 pin, then this is q 3. So, what will happen is that your combinational logic will evaluate. So, since this time period is 10 nanosecond, this t clock is 10 nanosecond, so the half period from here till this point is 5 nanosecond. So, my d 2 will be appearing, let us say at this point, d 2 will be available at this point. So, this will be from here to here is 8 nanosecond and this time is 3 nanosecond. So, your d 2 will be available at this point, at this time stamp. So, now after that, so this is 2 nanosecond which is basically responsible for t combinational 2 and this is the delay of t combinational 1. So, if you can see is that your time is borrowed across half cycle, but your data is stable before the rising age of the latch 3. So, across stages it is not borrowing, but inside the half cycle it is borrowing, inside the half cycle it is borrowing. And you can tell me how much time is borrowed here? The time borrowed here is basically this 3 nanosecond, this t borrow in this case is basically 3 nanosecond.

So, this is basically the example for time borrowing across half cycle. Now we will go to the time borrowing across stages. So, what it is doing? It is doing the borrowing in this stage. So, it is doing the time borrowing in across the stages, this is the second one. If you can take the same example L1, L2, combinational logic 2, then you have a latch 3.

So, this is D1, this is Q1, D2, Q2, D3, Q3. This is phi 1, this is phi 1, this is phi 2. So, let us say this delay is 8 nanosecond, same example. Here it is let us say 4 nanosecond. So, same example, but only difference is that your combinational logic 2 is 4 nanosecond. And your t clock, t clock is 10 nanosecond, t non-overlap is basically 0 and t D2Q maximum is also 0. We assume this for simplicity. It may come in real applications. So, what we are doing here, we have a phi 1 which is drawn like this. Then the phi 2 is basically opposite phase. So, it will be drawn like this. So, now if you see this period is basically t clock is 10 nanosecond and this one is basically 5 nanosecond.

So, now the D2 basically the D2 will appear around 8 nanosecond, around 8 nanosecond from here to here is 8 nanosecond. Now the combinational logic 2 will start to work after 8 nanosecond. So, how much time it is taking? So, D3 will be available, D3 will be available after 4 nanosecond. So, it is taking 2 nanosecond from this side and 2 nanosecond from the across the clock edges.

So, this borrowing is called across stage borrowing. We are borrowing across the stages. So, in this case if you can summarize L2 is borrowing how much time? L2 is borrowing 3 nanosecond. L3 is borrowing 2 nanosecond. So, this is across the stages. So, this is an example of time borrowing across stages. Now we will take an example where we are doing the time borrowing. So, this is third case. If I go back this is the third case where we are doing the time borrowing in a loop or a cycle, this third case. So, what it says that loop may borrow internally but must complete within the cycle. You can borrow internally how many stages you have does not matter but it should reach the first latch before the set of time of that latch.

Now take this example. Here all the delay values are written here. Your delta 1 is 550 picosecond, delta 2 is 580 picosecond, delta 3 is 450 picosecond, delta 4 is 200 picosecond. So, in this case we assume that, okay, let us assume your t clock is basically 1000 picosecond, okay. We will assume that your t clock is 1000 picosecond. And we also assume that t non-overlap is 0 and t d2q maximum is also 0. These are the assumptions. So, if this is the case how much time each of the latches will borrow? Latch number, amount of time it is borrowing, okay. So, latch 1 will borrow this one minus t clock by 2. This 550 minus t clock by 2, t clock by 2 is 500. So, it will borrow 50 picosecond. So, delta 1 minus t clock by 2. Now L2, how much L2 will borrow? How much L2 will borrow? Since we have 50 picosecond is used, so 450 is already done by this same t clock by 2. So, what is remaining there? 450 should be minus from 580, okay. 450 should be minus from, so delta 2 minus 450. What is the amount? It is 130 picosecond. Now what about L3? How much L3 will borrow? So, now 130 picosecond is already borrowed by L2. So, you have 500 minus 130, okay. So, you have to do 500 minus 130. It is 370. So, 370 picosecond. So, now in the third stage, it will be minus 370 picosecond. So, it is 450 minus 370, 80 picosecond. So, it is 80 picosecond. And the L4, it is 500 minus 80. It is basically 420 picosecond. So, your delta 4, since your available time is 420 picosecond, but it is delta 4 is basically less than 420 picosecond. So, it is not borrowing anything, no borrowing in this case.

No borrowing in this case. Let us assume your t clock is 800 picosecond. Then your setup time, t setup is 25 picosecond and t non-overlap is 0. This is a new case we are doing, analyzing the same system with these conditions, okay. So, I am writing latch number, then amount of time it is borrowing, borrow time, okay. So, this latch 1, we have basically your delta 1 minus t clock by 2. So, t clock by 2 is 400 and this is 550. So, it is borrowing 150 picosecond. Now after it is borrowing 150, how much time left out? So, it is 400 minus 150, it is basically 250 picosecond. So, this much time is left out for me. So, your L2, okay, so L2 can do borrow, how much time it will borrow? It is do delta 2 minus 250 picosecond.

So, this comes out to be 330 picosecond. So, your delta 2 is 580, 580 minus 250, 330 picosecond. Now L3, after I do 330, so t by 2 is 400 picosecond, 330 if I subtract, it is comes out to be 70 picosecond. So, I have available time 70 picosecond and the amount of work need to be done is 450 picosecond, okay. So, it will borrow delta 3, delta 3 minus 70 picosecond. Delta 3 is 450 minus 70, 450 minus 70 is basically 380 picosecond. So, what is our requirement is that? Our requirement is that your data should be stable before the setup time requirement of the latch. So, our half clock period, t clock by 2 is 400 picosecond. So, this is important. And our setup time, t setup is basically 25 picosecond. So, your data should be available at least 375 picosecond, okay. So, our maximum t borrow, whatever we found out, should be less than equals to t clock by 2 minus setup time as my t non-overlap is 0. So, as per this requirement, it should be 375. As per this requirement, these two, it should come to be 375 picosecond can be borrowed. But the requirement is 380. So, your system will fail. System will not satisfy the requirement when your t clock is 800 picosecond and t setup is 25 picosecond. So, we will discuss about the time borrowing in case of a pulsed latch. So, pulsed latch is like a normal latch but the width of the pulse is narrow, okay. So, during the narrow window, it is transparent, the latch is transparent. So, in case of a pulsed latch, your pulse width is narrow and it capture and it is transparent during that on period.

So, this is the concept of a pulsed latch. So, there are two cases are possible in case of a pulsed latch for time borrowing. So, what is that? Case 1 where your t pulse width, width of the pulse is greater than t setup, okay. So, this is case 1 where this is my pulse width, this is my pulse width and my setup time is something less than that, okay. Let us say this is t setup and this is my t pulse width. In that case, what is my amount of time borrowing? So, amount of time borrowing will be this much.

So, this is t borrow. So, t borrow, so that whatever I showed is the maximum, should be less than equals to t pulse width minus t setup, okay. So, this is case 1. Then the case 2 is that when my t pulse width is less than t setup. So, in that case, what is the conditions? Your data should be stable before the rising edge of the pulse. So, in this case, what is the condition? Let us say I have a pulse width is here and setup time of the latch is more than the pulse width. Let us say this is my t setup and this is my t pw, okay. Your t setup is greater than my t pulse width. In this case, your data should be stable before the rising edge of the pulse. So, in this case, there is no time borrowing is possible. So, no time borrowing is possible in this case. So, in this lecture, we discussed about the time borrowing in latches.

Thank you for your attention.

Thank you.